

UNIVERSITETET I OSLO  
Fysisk institutt

TNS med lavkostsensorer

Masteroppgave

Diego Mugisha

20.desember 2012





# Forord

Denne oppgaven er innlevert til institutt for fysikk ved Universitetet i Oslo som en del av min Mastergrad (M.Sc.). Denne masteroppgaven ble utført ved Universitetsenteret på kjeller (UNIK). Jeg har underveis i studiet spesialisert meg innefor Kybernetikk.

Jeg vil benytte denne anledningen til takke min veileder Professor Oddvar Hallingstad ved UNIK for mange kreative innspill og givende faglige diskusjoner i arbeidet med denne oppgaven.

Arbeidet med denne oppgaven har vært lærerikt på mange måter, og jeg er takknemlig for det.

Kjeller, 20. desember 2012

---

Diego Mugisha

Universitetet i Oslo



# Sammendrag

I denne masteroppgaven har det først laget en deterministisk banegenerator som beveger seg langs en firkant i xy-planet som genererer posisjon, hastighet, akselerasjon, spesifikk kraft og vinkelhastighet.

Micro- Electro- Mechanical System (MEMS) sensorer, av type gyro og akselerometer, har gjennom årene blitt et stort forskningsfelt med en rekke kommersielle suksesser. Det har blitt laget feilmodeller for treaksede akselerometer og gyroer, og har simulert sensorfeilene og laget de til de deterministiske sensormålingene funnet i banegeneratoren. Det har også blitt laget et program som løser navigasjonslikningene ved hjelp av integrasjonsalgoritmer, Euler's og Heun's metode. Dette blir sammenliknet med tilstandene fra den deterministiske banegeneratoren.

For å forsøke på å få en navigasjonsnøyaktigheten funnet ved løsning av navigasjonslikningene har det laget et kalmanfilter som estimerer borte store deler av feilen.



# Innhold

## Nomenklatur, notasjon

## 1 Innledning

1.1 Om oppgaven.....	19
1.2 Struktur.....	20

## 2 Matematisk grunnlag

2.1 Vektorer, matriser og koordinatsystemer.....	23
2.1.1 Vektorer.....	23
2.1.2 Skjevsymmetrisk form.....	24
2.1.3 Koordinatsystemer.....	24
2.2 Eulervinkler.....	25
2.2.1 3-2-1 Eulervinkler.....	25
2.2.2 Derivasjon av RKM.....	26
2.3 Støymodeller.....	28
2.3.1 Tilfeldig konstant.....	28
2.3.2 Browns bevegelse.....	28
2.3.3 Farget støy.....	28
2.4 Diskretisering av kontinuerlig prosess modell.....	29
2.4.1 Diskretisering av generell lineær prosess modell.....	29
2.4.2 Diskretisering av støymodell.....	32
2.5 Kalmanfilter.....	35
2.5.1 Lineært system for Kontinuerlig-diskret Kalmanfilter.....	35
2.5.2 Kalmanfilteret brukt på ulineært system.....	37

## 3 Gyro og akselerometer modeller

3.1 Introduksjon.....	39
3.2 Modell av gyroskopet.....	40

3.3	Modell av akselerometer.....	42
<b>4</b>	<b>Generering av gyro-og akselerometer målingene</b>	
4.1	Deterministisk banegenerator.....	45
4.1.1	Legemets bane.....	45
4.1.2	Simulertsresultat fra banegenerator.....	50
4.2	Stokastiske gyro- og akselerometer målinger.....	53
4.2.1	Simuleringsresultat for målefeil.....	57
<b>5</b>	<b>TNS</b>	
5.1	Tre-akset plattform med flat ikke-roterende jord.....	63
5.1.1	Sann ulineær systemmodell.....	63
5.1.2	Navigasjonslikninger.....	66
5.2	Integrasjonsrutine.....	67
5.3	Oppsummering av TNS'et.....	70
5.4	Simuleringsresultat for TNS.....	71
5.4.1	Simuleringsresultat for TNS ved bruk av Euler's metode.....	71
5.4.2	Simuleringsresultat for TNS ved bruk av Heun's metode.....	72
<b>6</b>	<b>Kalmanfilter for TNS</b>	
6.1	Linearisert Kalmanfilter (LKF).....	73
6.1.2	Simuleringsresultat for Kalmanfilteret.....	76
<b>7</b>	<b>Konklusjon.....</b>	<b>81</b>
<b>8</b>	<b>Videre arbeid.....</b>	<b>83</b>
	<b>Bibliografi.....</b>	<b>85</b>
<b>A</b>	<b>MATLAB-kode.....</b>	<b>87</b>



# Figurer

Figur 2.1: Blokkdiagram for random konstant.....	33
Figur 2.2: Blokkdiagram for browns bevegelse.....	33
Figur 2.3: Blokkdiagram for både tilfeldig konstant og browns bevegelse.....	33
Figur 2.4: Blokkskjema av et diskret kalmanfilter.....	36
Figur 3.1: Eksempel for Smartphone med sensor.....	40
Figur 4.1: Banegenerator-omløp.....	45
Figur 4.2: Simuleringsresultat for banegenerator.....	49
Figur 4.3: Simuleringsresultat for akselerasjon.....	50
Figur 4.4: Simuleringsresultat for hastighet.....	51
Figur 4.5: Simuleringsresultat for posisjon.....	51
Figur 4.6: Simuleringsresultat for vinkelhastighet.....	52
Figur 4.7: Simulert spesifikk kraft i x- og y-retninger.....	52
Figur 4.8: Simulert spesifikk kraft i z-retning.....	52
Figur 4.9: Simulert utvikling av støybidrag fra browns bevegelse for gyro.....	57
Figur 4.10: Simulert utvikling av støybidrag fra farget støy for gyro.....	57
Figur 4.11: Simulert utvikling av browns bevegelse for akselerometer.....	57
Figur 4.12: Simulert utvikling av farget støy for akselerometer.....	57
Figur 4.13: Målt spesifikk kraft om x-retning.....	58
Figur 4.14: Målt spesifikk kraft om y-retning.....	58
Figur 4.15: Målt spesifikk kraft om z-retning.....	58
Figur 4.16: Målt vinkelhastighet om x-retning.....	59

Figur 4.17: Målt vinkelhastighet om y-retning.....	59
Figur 4.18: Målt vinkelhastighet om z-retning.....	59
Figur 5.1: Diskretisering ved Heun's metode basert på Euler's metode.....	68
Figur 5.2: Oppsummering av Tregghetsnavigasjonssystemet.....	70
Figur 5.3: Posisjon for navigasjon ved bruk av Euler's metode.....	71
Figur 5.4: Hastighet for navigasjon ved bruk av Euler's metode.....	71
Figur 5.5: Avvik mellom posisjon og posisjon for navigasjon.....	71
Figur 5.6: Avvik mellom simulert hastighet og hastighet for navigasjon.....	71
Figur 5.7: Posisjon for navigasjon ved bruk av Heun's metode.....	72
Figur 5.8: Avvik mellom posisjon og posisjon for TNS.....	72
Figur 5.9: Hastighet for TNS ved bruk av Heun's metode.....	72
Figur 5.10: Avvik mellom hastighet og Hastighet for TNS.....	72
Figur 6.1: Blokkskjema av linearisert kalmanfilter.....	75
Figur 6.2: Estimert posisjon i x-retning med standardavvik.....	76
Figur 6.3: Estimert posisjon i y-retning med standardavvik.....	76
Figur 6.4: Estimert posisjon i z-retning med standardavvik.....	76
Figur 6.5: Estimert hastighet i x-retning med standardavvik.....	77
Figur 6.6: Estimert hastighet i y-retning med standardavvik.....	77

Figur 6.7: Estimert hastighet i z-retning med standardavvik.....	77
Figur 6.8: Estimert skjevstilling i x-retning.....	78
Figur 6.9: Estimert skjevstilling i y-retning.....	78
Figur 6.10: Estimert skjevstilling i z-retning.....	78

# Tabeller

Tabell 1: Forkoteleser.....	13
Tabell 2: Symbolbeskrivelse for kapittel 2.....	13
Tabell 3: Symbolbeskrivelse for kontinuerlig systemmodell.....	14
Tabell 4: Symbolbeskrivelse for diskret systemmodell.....	15
Tabell 5: Symbolbeskrivelse for en determinisk banegenerator.....	15
Tabell 6: Symbolbeskrivelse for målefeil.....	16
Tabell 7: Feildefinisjoner til tre-akset plattform.....	63
Tabell 8: Tre-akset plattform med ikke roterende jord.....	63

## Nomenklatur – symboler og forkortelser

Forkortelse	Betydning
KF	Kalmanfilter
TNS	Treghetsnavigasjonssystem
LKF	Lineær Kalmanfilter
IMU	Inertial Measure Unit
NAV	Navigasjon

Tabell 1: Forkortelser

Matematisk Symbol	Beskrivelse
$\vec{v}$	Geometrisk vektor
$\underline{v}$	Algebraisk vektor
$S(\omega)$	Skjevsymmetrisk form av $\omega$
$R_b^n$	Rotasjonsmatrisa fra b-ramma til n-ramma
$\Theta_1, \Theta_2, \Theta_3$	Eulervinklene Pitch, Roll og Yaw
$c_\Theta, s_\Theta$	Forkortelse for $\cos\Theta / \sin\Theta$
$\alpha$	Tilfeldig konstant
$\beta$	Browns bevegelse
$\gamma$	Farget støy

Tabell 2: Symbolbeskrivelse for kapittel 2

Symbol	Beskrivelse
$F(t)$	Systemmatrise
$L(t)$	Pådragsmatrise
$G(t)$	Prosesstøymatrise
$\underline{x}(t)$	Tilstandsvektor
$\underline{u}(t)$	Pådragsvektor
$\underline{z}(t)$	Målevektor
$\underline{v}(t)$	prosesstøy
$\underline{w}(t)$	Målestøy
$\tilde{Q}(t)$	Prosesstøyens spectraltetthet
$\tilde{R}(t)$	Målestøyens spektraltetthet
$\delta(t - \tau)$	Diracs deltafunksjon: $\int_{-\infty}^{\infty} f(\tau) \delta(t - \tau) d\tau = f(t)$

Tabell 3 : Symbolbeskrivelse for kontinuerlig systemmodell

Symbol	Beskrivelse
$k$	Tidspunkt $t_k$
$\underline{x}_k$	tilstandsvektor
$\underline{v}_k$	prosesstøy
$\underline{u}_k$	pådrag
$\underline{z}_k$	måling
$\underline{w}_k$	målestøy
$\Phi_k$	Systemmatrisa
$\Lambda_k$	pådragsmatrisa
$\Gamma_k$	prosesstøymatrisa
$H_k$	målematrisa
$P_0$	Initiell kovariansmatrise
$Q_k$	Prosesstøyens kovarians
$R_k$	Målestøysens kovarians
$\delta_{kl}$	Kronecker $\delta: \begin{cases} \delta_{kl} = 1 \text{ for } k = l \\ \delta_{kl} = 0 \text{ for } k \neq l \end{cases}$

Tabell 4 : Symbolbeskrivelse for diskret systemmodell

Matematisk Symbol	Beskrivelse
$\underline{p}^n$	Posisjonsvektor i n-ramma
$\underline{v}^n$	Hastighetsvektor i n-ramma
$\underline{a}^n$	Akselerasjonsvektor i n-ramma
$\underline{f}^n$	Vektor for spesifikk kraft i n-ramma
$\underline{\omega}_b^{nb}$	Vinkelhastigheten i b-ramma sett fra n-ramma representert i b-ramma
$p(t)$	Uttrykk for posisjon i banegenerator
$v(t)$	Uttrykk for hastighet i banegenerator
$a(t)$	Uttrykk for akselerasjon i banegenerator
$\underline{\Theta}$	Eulervinkler i banegenerator

Tabell 5: Symbolbeskrivelse for en determinisk banegenerator

Matematisk symbol	Beskrivelse
$\underline{x}^a$	Tilstandsvektor til prosessstøyen i akselerometer
$\underline{x}^g$	Tilstandsvektor til prosessstøyen i gyro
$\Phi^a$	Diskret systemmatrise i akselerometer
$\Gamma^a$	Diskret prosessstøymatrise i akselerometer
$\Phi^g$	Diskret systemmatrise i gyro
$\Gamma^g$	Diskret prosessstøymatrise i gyro
$\tilde{f}^b$	Målt spesifikk kraft
$\tilde{\omega}_{nb}^b$	Målt vinkelhastighet
$P_a$	Stasjonær kovariansmatrisen til akselerometer
$P_g$	Stasjonær kovariansmatrisen til gyro
$T_a$	Tidskonstanten til akselerometer
$T_g$	Tidskonstanten til gyro
$\underline{\alpha}^a$	Tilfeldig konstant i akselerometer
$\underline{\alpha}^g$	Tilfeldig konstant i gyro
$\underline{\beta}^a$	Browns bevegelse i aks.
$\underline{\beta}^g$	Browns bevegelse i gyro
$\underline{\gamma}^a$	Farget støy i aks.
$\underline{\gamma}^g$	Farget støy i gyro
$Q_{\gamma}^a$	Kovariansmatrise til prosessstøyen for farget støy, akselerometer
$Q_{\beta}^a$	Kovariansmatrise til prosessstøyen for browns bevegelse, akselerometer
$Q_{\gamma}^g$	Kovariansmatrise til prosessstøyen for farget støy, gyro
$Q_{\beta}^g$	Kovariansmatrise til prosessstøyen for browns bevegelse, gyro

Tabell 6: Symbolbeskrivelse for målefeil







# Kapittel 1

## Innledning

### 1.1 Om oppgaven

Masteroppgaven er, som nevnt i forordet, gjennomført i samarbeid med UNIK. Oppgaveteksten lyder som følger:

Det finnes nå tilgjengelig lavkost treaksede aksellerometer og gyroer. For å vurdere hvilken navigasjonsnøyaktighet disse sensorene kan gi det skal lages et simuleringsprogram for et treghetsnavigassystem (TNS) basert på disse sensorene. Trinnene for å lage simulatoren er:

1. Lag en deterministisk banegenerator som gir ut spesifikk kraft og vinkelhastighet for en treghetsplattform referert til plattformens koordinatsystem. La plattformen bevege seg langs en firkant i xy-planet.
2. Lag feilmodeller for treaksede aksellerometer og gyroer. Feilkildene kan være tilfeldig konstant, brownsk bevegelse og farget støy. Simuler sensorfeilene og legg de til de deterministiske sensormålingene funnet i banegeneratoren.
3. Lag et program som tar inn de genererte sensordataene og løser navigasjonslikningene.
4. For å forbedre navigasjonsnøyaktigheten funnet ved løsning av navigasjonslikningene skal det lages et kalmanfilter som utnytter kunnskapen om støykildene.
5. Simulatoren skal kjøres med typiske data for lavkost treghetskomponenter.

Referanser:

1. Jørn Skarbø Grahn: Estimering av MEMS-gyroparameter, Msc-avhandling UiO 20110629
2. Paal Alexander Nerholm: MEMS Inertial Navigation System, Msc-avhandling UiO 20111219
3. Hågen Kyllø: Treghetsnavigasjon med Smarttelefon, MSc-avhandling UiO 20120611

## 1.2 Struktur

Denne oppgaven er delt inn i 7 kapitler. Under følger en liten oversikt over innholdet i de forskjellige delene.

### **Kapittel 2: Matematisk bakgrunnstoff**

Denne delen består av en del matematisk bakgrunnstoff som kan være nyttig å kunne for å lese rapporten.

### **Kapittel 3: Gyro og aksellerometer modeller**

I dette kapitlet har det blitt analysert forskjellige støybidrag og satt opp en modell for målingene til gyroen og akselerometeret.

### **Kapittel 4: Banegenerator og feilmodellene**

I dette kapitlet har det blitt laget banegenerator og simuleringsmodell som generer målinger.

### **Kapittel 5: Treghetsnavigasjonssystem**

De matematiske modellene utledes for navigasjonslikningene.

### **Kapittel 6: Kalmanfilter for TNS**

Sammedrag av likninger for lineært kalmanfilter utdeles.

### **Kapittel 7: Konklusjon**

Konklusjon av oppgaver

### **Kapittel 8: Videre arbeid**

Anbefaling for videre arbeid.





# Kapittel 2

## Matematisk bakgrunnstoff

### 2.1 Vektorer og matriser

Vektor- og matriseberegninger [16] brukes generelt for å finne avstander, retninger og oppførselen til et legeme når man ser bort fra fysiske krefter som har en påvirkning.

Om legemet er i lufta, på bakken eller under vann har ingen betydning, samme matematiske formler og beskrivelser kan benyttes over store deler hvor fysiske lover er relevant .

#### 2.1.1 Vektorer

En vektor er en størrelse som har både lengde og retning. En vektor kan beskrives som geometrisk eller algebraisk. Gitt tre-dimensjonale koordinatsystem med basisvektorer  $\vec{b}_i$ , den geometriske vektor kan bli representert som en lineær kombinasjon av grunnlagsdata, som er gitt med vektor i rommet  $\mathcal{R}^n$ :

$$\vec{v} = \sum_{i=1}^3 v_i \vec{b}_i \quad (2.1)$$

hvor vektoren  $\vec{v}$  algebraisk kan uttrykkes som en matrise:

$$\underline{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (2.2)$$

#### 2.1.5 Skjevsymmetrisk form

En skjevsymmetrisk matrise  $A$  er definert slik at  $A = -A^T$ , og alle elementene langs hoveddiagonalen er lik 0. Den skjevsymmetriske form finnes ved å ta kryssproduktet av

$$s_{ij}^p = \langle (\omega_1 \vec{P}_1 + \omega_2 \vec{P}_2 + \omega_3 \vec{P}_3) \times \vec{P}_j, \vec{P}_i \rangle \quad (2.3)$$

hvor  $P$  er den basisvektoren som danner det ortogonale koordinatsystemet.

Den skjevsymmetriske formen til en vektor blir da

$$S(\underline{\omega}^p) = \underline{\omega}^p X = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \text{ hvor}$$

$$\vec{\omega} = \omega_1 \vec{P}_1 + \omega_2 \vec{P}_2 + \omega_3 \vec{P}_3 \quad (2.4)$$

### 2.1.3 Koordinatsystemer

I et treghetsnavigasjonssystem er det behov for et stor antall koordinatsystemer [16]. De forskjellige koordinatsystemene er treghetsramme (inertial), Jordramme (earth), Geografisk ramme, legemeramme (body), plattformramme og akselerometerramme, gyroramme.

I denne delen av koordinatsystemer blir presentert og definert av rammene.

#### Treghetsramme $\{i\}$

Koordinatsystemet til treghetsrammen kalles ECI som står for Earth-Centered Inertial.  $i$  brukes som symbol for treghetsramme. Aksenotasjon er  $x_i$ ,  $y_i$  og  $z_i$ .

For treghetsnavigasjonssystem som bruker akselerometer som er dårliger enn  $10^{-7} g$  kan vi bruke et kartesisk koordinatsystem med origo i jordas sentrum og som ikke roterer i forhold til fiksstjernene som treghetsramme. Vi lar videre  $z_i$ -aksen falle sammen med jordas rotasjonsakse og peke mot nordpolen.

#### Jordramme $\{e\}$

Koordinatsystemet ECEF står for Earth Centered Earth Fixed.  $e$  brukes som symbol for jordramme. Aksenavn er  $x_e$ ,  $y_e$  og  $z_e$ .

Jordramme er et kartesisk koordinatsystemet som ligger fast i jorda og med origo i jordas sentrum.

#### Legemeramme $\{b\}$

Legemeramme er en ramme ligger fast i legemet (vanligvis med origo i massesenteret).  $b$  brukes som suffiks for akselerometerramme. Aksenotasjon er  $x_b$ ,  $y_b$  og  $z_b$ .

#### Geografisk ramme $\{n\}$

$n$  brukes som suffiks for geografisk ramme. Geografisk ramme er en lokal navigasjonsramme som har sitt origo i systemet (massesenter, treghetsplattform), og



aksene peker nordover, østover og ned (vinkelrett på referanse ellipoiden).  $n$  brukes som suffiks for akselerometerramme. Aksenotasjon er  $x_n, y_n$  og  $z_n$ . Vi antar her at geografisk ramme faller sammen med treghetsramme:  $\{n\} = \{i\}$

### Akselerometerramme $\{a\}$

Vi har tre akselerometer som ligger langs hver sin akse i et kartesisk koordinatsystem.  $a$  brukes som suffiks for akselerometerramme. Aksenotasjon er  $x_a, y_a$  og  $z_a$ . Vi antar her at akselerometerramma faller sammen med legemeramme:  $\{a\} = \{b\}$

### Gyroramme $\{g\}$

Vi har tre gyroer som ligger langs hver sin akse i et kartesisk koordinatsystem.  $g$  brukes som symbol. Aksenotasjon er  $x_g, y_g$  og  $z_g$ . Vi antar her at akselerometerramma faller sammen med legemeramme:  $\{g\} = \{b\}$

## 2.2 Eulervinkler

### 2.2.1 3-2-1 Eulervinkler

Vektorene i denne oppgaven er definert i flere forskjellige rammer [16]. For å transformere mellom rammene innføres lineære operasjoner. Dette er operasjoner som transformere mellom vektorer og benevnes  $C_b^a$ . En operator som foretar en rotasjon om en av basisvektorene benevnes med  $R_b^a(\Theta)$  og kalles ofte retningskosinmatrise. Vi har 3 elementære retningskosinmatriser som definerer rotasjonen  $\Theta$  om basisvektor 1, 2 og 3:

$$R_1(\Theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\Theta & -s_\Theta \\ 0 & s_\Theta & c_\Theta \end{bmatrix} \quad R_2(\Theta) = \begin{bmatrix} c_\Theta & 0 & s_\Theta \\ 0 & 1 & 0 \\ -s_\Theta & 0 & c_\Theta \end{bmatrix} \quad R_3(\Theta) = \begin{bmatrix} c_\Theta & -s_\Theta & 0 \\ s_\Theta & c_\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Ved rotasjoner om alle 3 basisvektorene (aksene) vil de to rammene kunne oppnå enhver stilling rommet i forhold til hverandre. Retningskosinmatrisen blir da:

$$R_b^n = R_b^n(\Theta_1, \Theta_2, \Theta_3) = R_3(\Theta_3)R_2(\Theta_2)R_1(\Theta_1) = R_{321}(\underline{\Theta}) \quad (2.6)$$

Vinklene har her fått benevnningen  $\Theta_1$ ,  $\Theta_2$  og  $\Theta_3$ . Jeg vil i denne oppgaven kun benytte rotasjoner om nye akser og vinklene er da eulervinkler. Rotasjonen blir da følgende: Gitt systemet  $b$ . Vi roterer det en vinkel  $\Theta_1$  rundt  $x^b$ . Dette gir oss et nytt system  $b'$ . Vi roterer deretter en vinkel  $\Theta_2$  om  $y^{b'}$ . Dette gir oss systemet  $b''$ . Til sist roterer vi  $b''$  en vinkel,  $\Theta_3$  om  $x^{b''}$  og systemet  $n$  fremkommer. Vi har da transformert

fra b systemet via systemene  $b'$  og  $b''$  til systemet  $n$ . Denne rotasjons sekvensen kalles ofte **3-2-1 eulervinkler**.

Multipliseres de elementære RKM sammen får vi :

$$R_b^n(\Theta_1, \Theta_2, \Theta_3) = \begin{bmatrix} c_{\Theta_3} c_{\Theta_2} & c_{\Theta_3} s_{\Theta_2} s_{\Theta_1} - s_{\Theta_3} c_{\Theta_1} & c_{\Theta_3} s_{\Theta_2} c_{\Theta_1} + s_{\Theta_3} s_{\Theta_1} \\ s_{\Theta_3} c_{\Theta_2} & s_{\Theta_3} s_{\Theta_2} s_{\Theta_1} + c_{\Theta_3} c_{\Theta_1} & s_{\Theta_3} s_{\Theta_2} c_{\Theta_1} - c_{\Theta_3} s_{\Theta_1} \\ -s_{\Theta_2} & c_{\Theta_2} s_{\Theta_1} & c_{\Theta_2} c_{\Theta_1} \end{bmatrix}$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.7)$$

Vi har ovenfor brukt notasjonen:

- $n$  geografisk k.s med origo i legemet
- $b$  body k.s som ligger fast i legemet og har aksene R, P og Y (roll, pitch, yaw)
- $\Theta_1$  rullvinkel bruker ofte  $\varphi$  for denne
- $\Theta_2$  elevasjonsvinkel (pitch), bruker ofte  $\Theta$  for denne
- $\Theta_3$  asimutvinkel (yaw), bruker ofte  $\psi$  for denne

### Det inverse problem for 3-2-1 Eulervinklene :

Gitt  $R_b^n$  finn vinklene, har løsningen ( Craig 1989, s. 47)

$$\begin{aligned} \Theta_1 &= \text{atan2} \left( r_{32} / c_{\Theta_2}, r_{33} / c_{\Theta_2} \right) \\ \Theta_2 &= \text{atan2} \left( -r_{31}, \sqrt{r_{11}^2 + r_{21}^2} \right) \\ \Theta_3 &= \text{atan2} \left( r_{21} / c_{\Theta_2}, r_{11} / c_{\Theta_2} \right) \end{aligned} \quad (2.8)$$

Vi løser først for  $\Theta_2$ . For  $\Theta_2 = \pm 90^\circ$  har vi singularitet og bare summen av  $\Theta_1$  og  $\Theta_3$  kan beregnes.

## 2.2.2 Derivasjon av RKM

La  $R_b^n = R$ ,  $R$  er ortonormal RKM

Hvis  $R$  er ortonormal matrise  $R_b^n$  har vi:

$$RR^T = I \quad (2.9)$$

Derivasjon gir:

$$\dot{R}R^T + R\dot{R}^T = 0 \quad (2.10)$$

$S$  og  $S^T$  er definert slik:

$$S = \dot{R}R^T \text{ og } S^T = R\dot{R}^T \quad (2.11)$$

$S$  er en skjevsymmetrisk matrise

$$S = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (2.12)$$

Fra (2.10) etter å ha nullt med  $R$  får vi:

$$\dot{R} + R\dot{R}^T R = 0 \quad (2.13)$$

$$\dot{R} = -S^T R = SR, \quad -S^T = S \quad (2.14)$$

$$S = \dot{R}R^T \quad (2.15)$$

$$\dot{R} = SR \quad (2.16)$$

Den deriverte av en rotasjon matrise blir:

$$\dot{R} = S(\underline{\omega})R, \quad (2.17)$$

hvor  $\underline{\omega}$  er vinkelhastighet.

## 2.3 Støymodeller

### 2.3.1 Tilfeldig konstant

I denne oppgaven skal vi ikke bruke tilfeldig konstant fordi browns bevegelse inneholder både tilfeldig konstant og browns bevegelse [2]. Tilfeldig konstant kaller vi for  $\alpha$ ,

$$\text{der } \dot{\alpha} = 0 \quad (2.18)$$

### 2.3.2 Browns bevegelse

Browns bevegelse er en feil av samme type som angular random walk, men her snakker vi om en hvitstøy som gjør seg gjeldende på vinkelakselerasjons- nivå. Når vi måler for eksempel hastigheten blir denne hvitstøyen integrert opp og fører til en random walk prosess som legges på hastighetsmålingene. Matematisk kan denne støyen ses på som prosesstøy. Utviklingen til denne støyprosessen på hastighetsnivå kan beskrives med differensiallikningen:

$$\dot{\beta} = v \quad (2.19)$$

Der  $v_{\beta}(t)$  er hvit støy. I denne rapporten kaller vi browns bevegelse for  $\beta$ .

### 2.3.3 Farget støy

Målingene påvirkes også av farget støy. Gauss- markovprosess er en samlebetegnelse for spesielle random-prosesser som oppstår ved at hvit støy blir filtrert gjennom enkle filtre. Den kontinuerlige prosessen  $x(t)$  er en 1. ordens markovprosess for alle  $k$  og  $t_1 < t_2 < \dots < t_k$  når

$$F[x(t_k) | x(t_{k-1}), \dots, x(t_1)] = F[x(t_k) | x(t_{k-1})] \quad (2.20)$$

Det vil si at sannynlighetstetthetsfunksjonen for prosessen  $x(t_k)$  kun avhenger av verdien et tidsskritt tilbake i tid,  $x(t_{k-1})$ . Hvis den kontinuerlige prosessen  $x(t)$  er en 1.ordens markov, kan den beskrives av differensiallikningen,

$$\frac{dx}{dt} = -\frac{1}{T}x + v \quad (2.21)$$

hvor  $T$  er korrelasjonstiden og  $v$  er en hvit støy. Hvis vi har at sannsynlighetstetthetsfunksjonene til  $x$  og  $v$  er gaussiske, kalles dette en gauss-markov prosess.

I vårt tilfelle kaller vi den fargede støyen for  $\gamma$ , og vi kan nå sette opp differensiallikningen for den fargede støyen:

$$\dot{\gamma} = -\frac{1}{T}\gamma + v \quad (2.22)$$

## 2.4 Diskretisering av kontinuerlige prosess modeller

Under implementering i Matlab skal alt være på diskret form.

### 2.4.1 Diskretisering av generell prosess modell

Discretisering er en prosess som overfører kontinuerlig tid modeller i diskrete modeller [17].

Gitt systemet:

$$\begin{aligned} \dot{\underline{x}}(t) &= F(t)\underline{x}(t) + L(t)\underline{u}(t) + G(t); \\ \underline{x}_0 &\sim N(\bar{\underline{x}}_0, \bar{P}_0), \underline{v} \sim N(\underline{0}, \tilde{Q}\delta(t-\tau)) \end{aligned} \quad (2.23)$$

Diskretisering kan skrives som:

$$\underline{\dot{x}}_{k+1} = \Phi_k \underline{x}_k + \Lambda_k \underline{u}_k + \Gamma_k \underline{v}_k; \underline{x}_0 \sim N(\bar{\underline{x}}_0, \bar{P}_0), \underline{v}_k \sim N(\underline{0}, Q\delta_{kl}) \quad (2.24)$$

Krever at

$$\bar{\underline{x}}(t_k) = \bar{\underline{x}}_k, \bar{P}(t_k) = \bar{P}_k \quad (2.25)$$

hvor

$$\Phi_k = e^{F((k+1)-k)} \quad (2.26)$$

$$\Lambda_k = \int_k^{k+1} e^{F((k+1)-\tau)} L_k \underline{u}(\tau) d\tau \quad (2.27)$$

$$\Gamma_k = \int_k^{k+1} e^{F((k+1)-\tau)} G_k \underline{v}(\tau) d\tau \quad (2.28)$$

Siden navigasjon ligningene omfatter ikke  $L(t)\underline{u}(t) / \Lambda_k$  anses å være null.

$$\dot{P} = FP + PF^T + G\tilde{Q}G^T, P(t_0) \text{ gitt} \quad (2.29)$$

Vi antar at F, G og  $\tilde{Q}$  er tidsinvariante.

Ved  $t = t_0$ :  $X(t_0) = P(t_0)$ , det vil si at  $Z(t_0) = I$   
hvor I er identitetsmatrise.

Definerer at :

$$P(t) = X(t) \cdot Z^{-1}(t) \quad (2.30)$$

Likning (2.30) kan skrives  $X = PZ$  og deriveres får vi:

$$\dot{X} = \dot{P}Z + P\dot{Z} \quad (2.31)$$

Vi har en differensiallikning for  $\dot{P}$  som setter inn:

$$\begin{aligned} \dot{X} &= (FP + PF^T + G\tilde{Q}G^T)Z + P\dot{Z} \\ &= FPZ + PF^TZ + G\tilde{Q}G^T + P\dot{Z} \end{aligned} \quad (2.32)$$

$$\begin{aligned} \dot{X} - FX - G\tilde{Q}G^T &= PF^TZ + P\dot{Z} \\ &= P(F^TZ + \dot{Z}) \end{aligned} \quad (2.33)$$

Dette kan skrives :

$$\begin{bmatrix} \dot{X} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} F & G\tilde{Q}G^T \\ 0 & -F^T \end{bmatrix} \begin{bmatrix} X \\ Z \end{bmatrix}, \begin{bmatrix} X(t_0) \\ Z(t_0) \end{bmatrix} \text{ hvor } X(t_0) = P_0 \text{ og } Z(t_0) = I \quad (2.34)$$

Løsningen av (2.34) er [17] :

$$\begin{aligned} \Phi_{xz}(t, t_0) &= e^{F_{xz}(t-t_0)} \\ &= \sum_{i=0}^{\infty} \frac{1}{i!} F_{xz}^i (t-t_0)^i \\ &= \begin{bmatrix} \Phi_{11}(t, t_0) & \Phi_{12}(t, t_0) \\ 0 & \Phi_{22}(t, t_0) \end{bmatrix} \end{aligned} \quad (2.35)$$

$$\begin{bmatrix} X(t) \\ Z(t) \end{bmatrix} = \begin{bmatrix} \Phi_{11}(t, t_0) & \Phi_{12}(t, t_0) \\ 0 & \Phi_{22}(t, t_0) \end{bmatrix} \begin{bmatrix} P_0 \\ I \end{bmatrix} \quad (2.36)$$

Ligningene blir:

$$Z(t) = \Phi_{22}(t, t_0), \quad X(t) = \Phi_{11}(t, t_0)P_0 + \Phi_{12}(t, t_0)I \quad (2.37)$$

Sette den i (2.30) får vi:

$$\begin{aligned} P(t) &= [\Phi_{11}(t, t_0)P_0 + \Phi_{12}(t, t_0)]\Phi_{22}^{-1}(t, t_0) \\ &= \Phi_{11}(t, t_0)P_0\Phi_{22}^{-1}(t, t_0) + \Phi_{12}(t, t_0)\Phi_{22}^{-1}(t, t_0) \end{aligned} \quad (2.38)$$

$$\text{Her er } \Phi_{12}(t, t_0)\Phi_{22}^{-1}(t, t_0) = \Gamma\tilde{Q}\Gamma^T \quad (2.40)$$

$$\bar{P}(t_{k+1}) = \Phi(t_{k+1}, t_k)\bar{P}(t_k)\Phi(t_{k+1}, t_k) + \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)G\tilde{Q}G^T\Phi(t_{k+1}, \tau)d\tau \quad (2.41)$$

$$\bar{P}_{k+1} = \Phi\bar{P}_k\Phi^T + \Gamma Q\Gamma^T \quad (2.42)$$

Ved å sammenlikne likningene og sette  $\Phi(t_{k+1}, t_k) = e^{F(t_{k+1}-t_k)}$  får vi

$$\Phi = e^{F(t_{k+1}-t_k)} = e^{F\Delta t} \quad (2.43)$$

$$\Lambda \underline{\underline{u}}_k = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)L\underline{\underline{u}}(\tau)d\tau \quad (2.44)$$

Dersom antar at  $\underline{\underline{u}}(\tau) = \underline{\underline{u}}_k$  for  $\tau \in [t_k, t_{k+1}]$  så har vi

$$\begin{bmatrix} \dot{\underline{\underline{x}}} \\ \dot{\underline{\underline{u}}} \end{bmatrix} = \begin{bmatrix} F & L \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \underline{\underline{x}} \\ \underline{\underline{u}} \end{bmatrix}, \quad \begin{bmatrix} \underline{\underline{x}}(t_k) \\ \underline{\underline{u}}(t_k) \end{bmatrix} = \begin{bmatrix} \underline{\underline{x}}_k \\ \underline{\underline{u}}_k \end{bmatrix} \quad (2.45)$$

$$\begin{bmatrix} F & L \\ 0 & 0 \end{bmatrix} = \tilde{F} \quad (2.46)$$

$$\begin{bmatrix} \underline{\underline{x}}(t) \\ \underline{\underline{u}}(t) \end{bmatrix} = e^{\tilde{F}(t-t_k)} \begin{bmatrix} \underline{\underline{x}}_k \\ \underline{\underline{u}}_k \end{bmatrix} \quad (2.47)$$

$$\tilde{\Phi} = \begin{bmatrix} \tilde{\Phi}_{11} & \tilde{\Phi}_{22} \\ 0 & I \end{bmatrix} \text{ for } t = t_{k+1} \quad (2.48)$$

$$\underline{\bar{x}}(t_{k+1}) = \tilde{\Phi}_{11} x_k + \tilde{\Phi}_{12} u_k \quad (2.49)$$

$$\Gamma Q \Gamma^T = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) G \tilde{Q} G^T \Phi^T(t_{k+1}, \tau) d\tau \quad (2.50)$$

Fant vi også at

$$\Gamma Q \Gamma^T = \tilde{\Phi}_{12}(t_{k+1}, t_k) \cdot \tilde{\Phi}_{22}(t_{k+1}, t_k) \quad (2.51)$$

hvor

$$\tilde{\Phi} = \begin{bmatrix} \tilde{\Phi}_{11} & \tilde{\Phi}_{12} \\ 0 & \tilde{\Phi}_{22} \end{bmatrix} = e^{\tilde{F} \cdot \Delta t} \quad (2.52)$$

$$\tilde{F} = \begin{bmatrix} F & G \tilde{Q} G^T \\ 0 & -F^T \end{bmatrix}, \Delta_t = t_{k+1} - t_k \quad (2.53)$$

## 2.4.2 Diskretisering av støymodell

Under implementering i Matlab må alt være på diskret form. Den kontinuerlige delen av støymodellen må diskretiseres og få den over på den diskrete formen [17].

### Tilfeldig konstant

Random konstant er beskrevet av vektor differensial ligningen

$$\dot{\alpha} = 0; \alpha_0 \sim N(\alpha_0, P_0) \quad (2.54)$$

$$\alpha(t) = \alpha_0, P(t) = P_0 \quad (2.55)$$

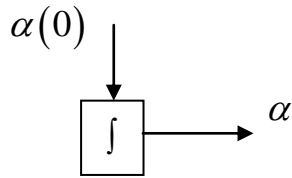
Den tilsvarende diskrete prosessen er beskrevet av

$$\alpha_{k+1} = \alpha_k, \alpha_0 \sim N(\alpha_0, P_0) \quad (2.56)$$

$$P_{k+1} = P_k \quad (2.57)$$

Stokastisk konstant kan være tenkt som utgang av integrator som har ingen inngang, men har en tilfeldig startbetingelsen.  $\alpha(0)$  er startbetingelsen.





Figur 2.1: Blokkdiagram for random konstant [2].

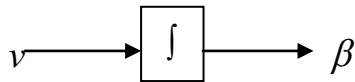
## Browns bevegelse

Browns bevegelse kommer når ukorrelerte signaler er integrert. Variabel differensial ligningen for browns bevegelse prosessen [17] er

$$\dot{\beta} = v \quad (2.58)$$

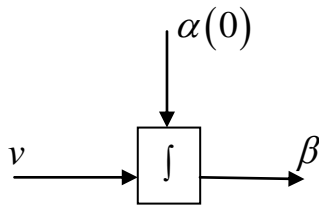
$$\beta(t) = \beta(t_0) + \int_{t_0}^t v(\tau) d\tau \quad (2.59)$$

$$\beta(t) = \beta_0, P(t) = P_0 + \tilde{Q}(t - t_0) \quad (2.60)$$



Figur 2.2: Blokkdiagram for Browns bevegelse [2].

En kombinasjon av tilfeldig konstant og browns bevegelse kan representeres ved bruk av bare en tilstand variabel.



Figur 2.3: Blokkdiagram for både tilfeldig konstant og browns bevegelse [2].

Tilsvarende diskrete prosessen er

$$\beta_{k+1} = \beta_k + v_k; \beta_0 \sim N(\beta_0, P_0) \quad (2.61)$$

$$v_k \sim N(0, Q)$$

$$\beta_k = \beta_0 + \sum_{i=0}^{k-1} v_i \quad (2.62)$$

$$\bar{\beta}_{k+1} = \bar{\beta}_k \dots = \bar{\beta}_0 \quad (2.63)$$

Ligningen for feilen kovariansen er

$$\bar{\mathbf{P}}_{k+1} = \Phi \mathbf{P}_k \Phi^T + \Gamma \mathbf{Q} \Gamma^T = \bar{\mathbf{P}}_k + \mathbf{Q} \quad (2.64)$$

$$\bar{\mathbf{P}}_k = \bar{\mathbf{P}}_0 + k \cdot \mathbf{Q} \quad (2.65)$$

### Farget støy

Variable differensiale ligninger for farget støy er

$$\begin{aligned} \dot{\gamma} &= -\frac{1}{T} \gamma + v; \gamma(t_0) \sim N(\bar{\gamma}_0, \bar{\mathbf{P}}_0), t_0 = 0 \\ v(t) &\sim N(0, \tilde{\mathbf{Q}} \cdot \delta(t - \tau)) \end{aligned} \quad (2.67)$$

$$\dot{\bar{\gamma}} = -\frac{1}{T} \bar{\gamma}, \bar{\gamma}(t_0) = \bar{\gamma}_0 \quad (2.68)$$

$$\bar{\gamma}(t) = e^{-\frac{t-t_0}{T}} \cdot \bar{\gamma}_0 \quad (2.69)$$

$$\dot{\bar{\mathbf{P}}} = -\frac{2}{T} \bar{\mathbf{P}} + \tilde{\mathbf{Q}}, \bar{\mathbf{P}}(t_0) = \bar{\mathbf{P}}_0 \quad (2.70)$$

$$\bar{\mathbf{P}}_\infty = \frac{\tilde{\mathbf{Q}} T}{2}, \quad (2.71)$$

(2.71) kan skrives slik

$$\tilde{\mathbf{Q}} = \frac{2\bar{\mathbf{P}}_\infty}{T} \quad (2.72)$$

Tilsvarende diskrete prosessen er

$$\gamma_{k+1} = \Phi \gamma_k + v_k; x_0 \sim N(\bar{x}_0, \bar{\mathbf{P}}_0) \quad (2.73)$$

$$E\{x_0, v_k\} = 0, v_k \sim N(0, \mathbf{Q} \zeta_{kj}) \quad (2.74)$$

$$\bar{\gamma}_{k+1} = \Phi \bar{\gamma}_k, \bar{\gamma}_0 \text{ gitt} \quad (2.75)$$

$$\bar{\mathbf{P}}_{k+1} = \Phi^2 \bar{\mathbf{P}}_k + \mathbf{Q}, \bar{\mathbf{P}}_0 \text{ gitt} \quad (2.76)$$

$$\bar{\mathbf{P}}_k = \Phi^{2k} \bar{\mathbf{P}}_0 + \sum_{i=0}^{k-1} \Phi^{2i} \mathbf{Q} \quad (2.77)$$

## 2.5 Kalmanfilter

### 2.5.1 Lineært system med Kontinuerlig- diskret Kalmanfilter

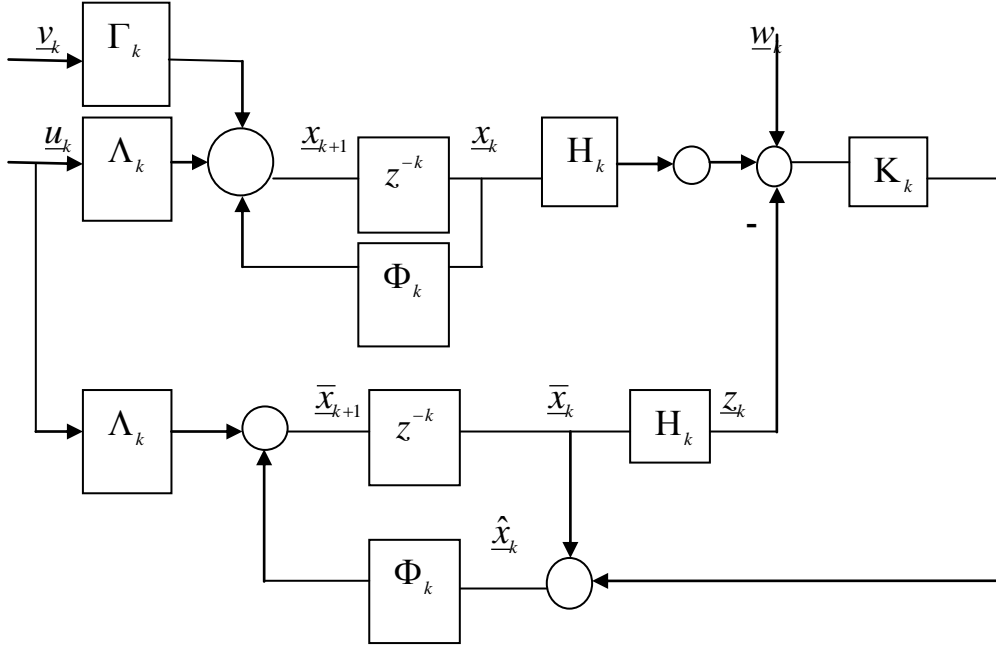
Kalmanfilteret er oppkalt etter Rudolf Emil Kalman som var en av de primære utviklerne av teorien. Et kalmanfilter er et sett av matematiske ligninger som implementerer en prediktor-korrektor estimator, som både er optimal og rekursiv som gir mulighet til å beregne tilstanden for en prosess basert på diskrete data påvirket av hvit støy [2]. Kalmanfilter minimerer gjennomsnitt av den kvadrerte feil. Filteret er veldig kraftig i flere aspekter: den støtter beregninger av fortid, nåtid og fremtids tilstander. En optimal estimator er en beregningsalgoritme som på grunnlag av kunnskap om prosess og sensorer, samt kunnskap om prosess- og målestøy kan beregne et minimum- varians estimat av tilstanden. Det vil si at vi ønsker å beskrive systemet gjennom en ordinær differensiallikning og en algebraisk likning [18]:

$$\dot{\underline{x}} = \underline{F}\underline{x} + \underline{G}\underline{v} + \underline{L}\underline{u} \quad (2.78)$$

$$\underline{z}_k = \underline{H}\underline{x}_k + \underline{w}_k \quad (2.79)$$

I tillegg trenger vi  $\underline{x}_0$ ,  $\underline{v}$  og  $\underline{w}$  beskrevet statistisk, samt at vi antar følgende betingelser:

$$\begin{aligned} E\{\underline{x}(t_0)\} &= \bar{\underline{x}}_0 \quad E\{(\underline{x}(t_0) - \bar{\underline{x}}_0)(\underline{x}(t_0) - \bar{\underline{x}}_0)^T\} = \underline{P}_0 \quad E\{\underline{x}(t_0)\underline{v}^T(t)\} = \underline{0} \\ E\{\underline{v}(t)\} &= \underline{0} \quad E\{\underline{v}(t)\underline{v}^T(\tau)\} = \delta(t - \tau)\tilde{\underline{Q}}(t) \quad E\{\underline{x}(t_0)\underline{w}_k^T\} = \underline{0} \\ E\{\underline{w}_k\} &= \underline{0} \quad E\{\underline{w}_k\underline{w}_l^T\} = \delta_{kl}\underline{R}_k \quad E\{\underline{v}(t)\underline{w}_k^T\} = \underline{0} \end{aligned}$$



Figur 2.4: Blokkskjema av et diskret Kalmanfilter

## Kontinuerlig-diskret Kalmanfilter

Kontinuerlig- diskret Kalmanfilter har følgende likninger:

Tidsoppdatering :

$$\dot{\underline{x}}(t) = F(t)\underline{x}(t) + L(t)\underline{u}(t), \underline{x}(t_k^+) := \hat{\underline{x}}; \quad \hat{\underline{x}}_0 \text{ gitt} \quad (2.80)$$

$$\dot{\bar{P}}(t) = F(t)\bar{P}(t) + \bar{P}(t)F^T + G(t)\tilde{Q}(t)G^T(t), \bar{P}(t_k^+) := \hat{P}_k; \hat{P}_0 \text{ gitt} \quad (2.81)$$

Måleoppdatering:

$$\hat{\underline{x}}_k = \bar{\underline{x}}_k + K_k (z_k - H_k \bar{\underline{x}}_k), \bar{\underline{x}}_k := \bar{\underline{x}}(t_k) \quad (2.82)$$

$$K_k = \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + R_k)^{-1}, \bar{P}_k := \bar{P}(t_k) \quad (2.83)$$

$$\hat{P}_k (I - K_k H_k) \bar{P}_k \quad (2.84)$$

## 2.5.2 Kalmanfilteret brukt på ulineært system

Når man har laget en matematisk modell av det fysiske systemet man ønsker å analysere, kan man sette opp filtermodellen og implementere Kalmanfilteret på en datamaskin. Kalmanfilteret jobber på en måte at det beregner en Kalmanfilterforsterkning, og utvikler middelerverdi og kovarians for hver tilstand over tid, slik at estimatet beregnes optimalt [15].

### Linearisert Kalmanfilter

Et linearisert Kalmanfilter er egentlig et "vanlig" lineært Kalmanfilter [18] som estimerer differansen mellom sann og beregnet måling. Denne differansen defineres slik:

$$\delta \underline{z}_k = \underline{z}_k - \tilde{\underline{z}}_k \quad (2.85)$$

$$\delta \underline{x}(t) = \underline{x}(t) - \tilde{\underline{x}}(t) \quad (2.86)$$

Første ordens approksimasjon av  $\delta \hat{\underline{x}}_k$  og  $\delta \underline{x}(t)$  gir følgende differensial-likninger for det lineariserte Kalmanfilteret:

$$\delta \hat{\underline{x}}_k = \delta \underline{\bar{x}}_k + \mathbf{K}_k^* (\delta \underline{z}_k - \mathbf{H}_k^* \delta \underline{\bar{x}}_k) \quad (2.87)$$

$$\delta \underline{z}_k = \delta \mathbf{H}_k \cdot \delta \underline{x}_k + \underline{w}_k \quad (2.88)$$

$$\delta \underline{x}(t_0) = \underline{x}(t_0) - \tilde{\underline{x}}(t_0) \quad (2.89)$$

Følgende likninger er gitt for det delvis tilbakekoblede Kalmanfilteret:

Måleoppdatering:

$$\delta \tilde{\underline{x}}_k = \delta \underline{\bar{x}}_k + \mathbf{K}_k^* (\delta \underline{z}_k - \mathbf{H}_k^* \delta \underline{\bar{x}}_k), \quad \delta \underline{\bar{x}} := \delta \underline{\bar{x}}(t_k) \quad (2.90)$$

$$\mathbf{K}_k^* = \bar{\mathbf{P}}_k \mathbf{H}_k^{*\top} (\mathbf{H}_k^* \bar{\mathbf{P}}_k \mathbf{H}_k^{*\top} + \mathbf{R}_k^*)^{-1}, \quad \bar{\mathbf{P}}_k := \bar{\mathbf{P}}(t_k) \quad (2.91)$$

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k \cdot \delta \mathbf{H}_k) \bar{\mathbf{P}}_k \quad (2.92)$$

Tidsoppdaring:

$$\delta \underline{\bar{x}}(t) = \mathbf{F}^*(t) \cdot \delta \underline{\bar{x}}(t); \quad \delta \underline{\bar{x}}(\hat{t}_k) = \delta \hat{\underline{x}}_k \quad (2.93)$$

$$\bar{\mathbf{P}}(t) = \mathbf{F}^*(t) \bar{\mathbf{P}}(t) + \bar{\mathbf{P}}(t) \mathbf{F}^{*\top} + \mathbf{G}^*(t) \tilde{\mathbf{Q}}^*(t) \mathbf{G}^{*\top}(t), \quad \bar{\mathbf{P}}(\hat{t}_k) := \hat{\mathbf{P}}_k \quad (2.94)$$

Estimatene for tilstandsvektoren beregnes da slik:

$$\underline{\hat{x}}(t_k) = \underline{\tilde{x}}_k + \delta \underline{\hat{x}}_k \quad (2.95)$$

$$\underline{\bar{x}}(t) = \underline{\tilde{x}}(t) + \delta \underline{\bar{x}}(t) \quad (2.96)$$

# Kapittel 3

## Gyroskoper og akselerometer modeller

### 3.1 Introduksjon

Et **akselerometer** [9] er en innretning for måling av akselerasjon og krefter induisert av tyngdekraft. Enkel- og fleraksemodeller er tilgjengelige for å detektere størrelsen og retningen på akselerasjonen som en vektorstørrelse. Akselerometere kan brukes for måle helning, vibrasjon og støt. De brukes i økende grad i bærbar elektronikk.

Et akselerometer måler akselerasjonen og tyngdekraften den gjennomgår. Begge måles i SI-enheter  $m / s^2$  eller  $g$  - krefter.

Moderne akselerometere er ofte små mikro-elektroniske systemer og er faktisk de enkelte MEMS-innretningene som finnes. De består stort sett av en frittstående stang med et lodd og en type elektronikk for måling av forskyvning. Under påvirkning av tyngdekraft eller akselerasjon blir loddet flyttet fra nøytralposisjonen. Forskyvningen måles analogt eller digitalt.

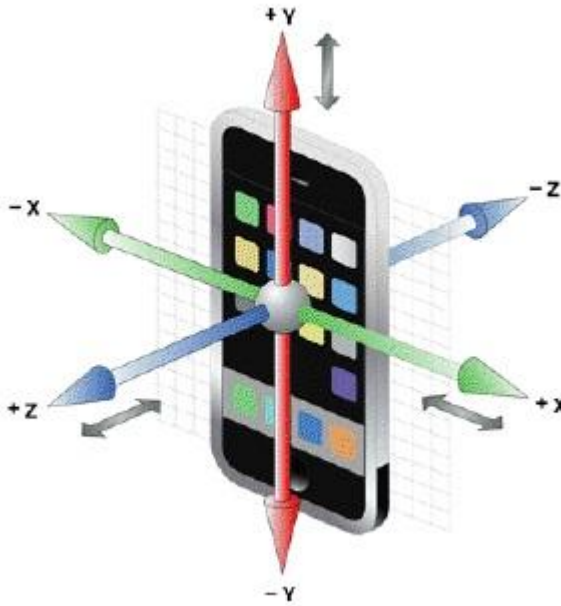
Enkelakse-, dobbelakse- og trippelaksemodeller eksisterer for å måle akselerasjon som en vektorstørrelse eller i en eller flere komponenter. MEMS-akselerometere er tilgjengelige for et vidt spekter av måleområder. Noen kan måle opp til tusener av  $g$ -krefter.

**Gyroskopet** [10] ble oppfunnet av Johan Gottlieb Friedrich von Bohnenberger i 1817 og navngitt av Jean Bernard Leon Foucault i 1892.

Et gyroskop måler vinkelhastigheten av systemet i treghetsreferanse rammen. Ved å bruke den opprinnelige orientering av systemet i treghetsreferanse rammen som startverdi og integrere vinkelhastigheten, er systemet nåværende orientering kjent til enhver tid.

En vanlig modell av et gyroskop består av et svinghjul med rotasjonsakse satt symmetrisk inn i flere ringer utenpå hverandre. Opphengt på den denne måten kan rotasjonsaksen fritt innta en hvilken som helst retning.

Akselerometer brukes sammen med gyroskop for navigering og styring. Akselerometer benyttes i håndkontroller for dataspill og i enkelte mobiltelefoner, blant annet for å skifte mellom portrettformat og landskapsformat ved visning av tekst og bilder.



Figur 3.1: Eksempel for Smartphone med sensor [7].

Akselerasjon måles på tre akser. Orientering sensor ( og koordinatsystem) varierer mellom ulike enheter.

I denne kapittelen skal sette opp matematisk modell av sensorene.

### 3.2 Modell av gyroskopet

Ut fra de feilkildene vi har sett på i kapittel 2, kan vi nå oppsummere **diskret modell for en akse gyro** for tilfeldig konstant, browns bevegelse og farget støy for en akset gyro:

$$\underline{\alpha}_{k+1}^g = \underline{\alpha}_k^g \quad (3.1)$$

$$\underline{\beta}_{k+1}^g = \underline{\beta}_k^g + \underline{v}_k^g \quad (3.2)$$

$$\underline{\gamma}_{k+1}^g = \Phi^g \underline{\gamma}_k^g + \underline{v}_k^g \quad (3.3)$$

$$\dot{\underline{P}}_{\beta}^g = \tilde{\underline{Q}}_{\beta}^g \underline{T} \quad (3.4)$$

$$\dot{\underline{P}}_{\gamma}^g = -\frac{2}{T} \underline{\bar{P}}_{\gamma}^g + \tilde{\underline{Q}}_{\gamma}^g \quad (3.5)$$

Stasjonært

$$\tilde{\underline{Q}}_{\gamma}^g = \frac{2}{T} \underline{\bar{P}}_{\gamma, \infty}^g \quad (3.6)$$



$$\text{NB: } \underline{\dot{\alpha}}_g = \underline{0} \text{ der } \underline{\alpha}_g \sim N(\underline{0}, \sigma_g^2 \mathbf{I}) \quad (3.7)$$

Under implementering i Matlab må alt være på diskret form. Den kontinuerlige delen av støymodellen må derfor diskretiseres og få den over på diskrete formen:

$$\Phi_\beta^g = \Phi_\beta^g(t_{k+1}, t_k) = e^{F_g \cdot \Delta t} \quad (3.8)$$

$$\tilde{Q}_\beta^g = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) G Q_\beta^g G^T \Phi(t_{k+1}, \tau)^T d\tau \quad (3.9)$$

$$\begin{aligned} \Gamma_\beta^g &= \Gamma_\beta^g Q_\beta^g (\Gamma_\beta^g)^T = \int_{t_k}^{t_{k+1}} \Phi_\beta^g G \tilde{Q}_\beta^g G^T (\Phi_\beta^g)^T \\ &= \tilde{\Phi}_{\beta,12}^g(t_{k+1}, t_k) \tilde{\Phi}_{\beta,22}^g(t_{k+1}, t_k) \end{aligned} \quad (3.10)$$

$F_g$  blir 0 siden det ikke er noen tilbakekobling på tilstanden browns bevegelse og  $G=1$ . Innsatt dette gir følgende verdier i den diskrete modellen:

$$\Phi_\beta^g = \Phi_\beta^g(t_{k+1}, t_k) = e^{0 \cdot (t_{k+1} - t_k)} = 1 \quad (3.11)$$

$$\begin{aligned} \tilde{Q}_\beta^g &= \int_{t_k}^{t_{k+1}} \Phi_\beta^g(t_{k+1}, \tau) G Q_\beta^g G^T \Phi_\beta^g(t_{k+1}, \tau)^T d\tau \\ &= \int_{t_k}^{t_{k+1}} Q_\beta^g d\tau = Q_\beta^g \cdot \Delta t \end{aligned} \quad (3.12)$$

Farget støy må også diskretiseres og få den over på diskrete formen:

$$\Phi_\gamma^g = \Phi_\gamma^g(t_{k+1}, t_k) = e^{F_g \cdot \Delta t} \quad (3.13)$$

$$\begin{aligned} \tilde{Q}_\gamma^g &= \int_{t_k}^{t_{k+1}} \Phi_\gamma^g(t_{k+1}, \tau) G Q_\gamma^g G^T \Phi_\gamma^g(t_{k+1}, \tau)^T d\tau \\ &= \int_{t_k}^{t_{k+1}} Q_\gamma^g \cdot d\tau = Q_\gamma^g \cdot \Delta t \end{aligned} \quad (3.14)$$

$$\begin{aligned} \Gamma_\gamma^g &= \Gamma_\gamma^g Q_\gamma^g (\Gamma_\gamma^g)^T = \int_{t_k}^{t_{k+1}} \Phi_\gamma^g(t_{k+1}, \tau) G Q_\gamma^g G^T \Phi_\gamma^g(t_{k+1}, \tau)^T d\tau \\ &= \Phi_{\gamma,12}^g(t_{k+1}, t_k) \Phi_{\gamma,22}^g(t_{k+1}, t_k) \end{aligned} \quad (3.15)$$

### 3.3 Modell av akselerometeret

Et akselerometer skal brukes for å måle spesikk kraft for en akset og en tre akset.

For **en akset akselerometer** er diskret modell definert som:

$$\underline{\alpha}_{k+1}^a = \underline{\alpha}_k^a \quad (3.16)$$

$$\underline{\beta}_{k+1}^a = \underline{\beta}_k^a + \underline{v}_k^a \quad (3.17)$$

$$\underline{\gamma}_{k+1}^a = \Phi^a \underline{\gamma}_k^a + \underline{v}_k^a \quad (3.18)$$

$$\dot{\underline{P}}_\beta^a = \tilde{Q}_\beta^a t \quad (3.19)$$

$$\dot{\underline{P}}_\gamma^a = -\frac{2}{T_a} \underline{\gamma}^a + \underline{v}_\gamma^a \quad (3.20)$$

Stasjonært

$$\tilde{Q}_\gamma^a = \frac{2}{T_a} \bar{\underline{P}}_{\gamma,\infty}^a \quad (3.21)$$

$$\textbf{NB: } \dot{\underline{\alpha}}^a = \underline{0} \text{ der } \underline{\alpha}^a \sim N(\underline{0}, \sigma_a^2 \cdot \mathbf{I}) \quad (3.22)$$

**Diskretisering av støymodellen:**

$$\Phi_\beta^a = \Phi_\beta^a(t_{k+1}, t_k) = e^{F_a \cdot \Delta t} \quad (3.23)$$

$$\tilde{Q}_\beta^a = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) G Q_\beta^a G^T \Phi(t_{k+1}, \tau)^T d\tau \quad (3.24)$$

$$\begin{aligned} \Gamma_\beta^a &= \Gamma_\beta^a Q_\beta^a (\Gamma_\beta^a)^T = \int_t^{t_{k+1}} \Phi_\beta^a G \tilde{Q}_\beta^a G^T (\Phi_\beta^a)^T \\ &= \tilde{\Phi}_{\beta,12}^a(t_{k+1}, t_k) \tilde{\Phi}_{\beta,22}^a(t_{k+1}, t_k) \end{aligned} \quad (3.25)$$

$F_a$  blir 0 siden det ikke er noen tilbakekobling på tilstanden browns bevegelse og  $G=1$ . Innsatt dette gir følgende verdier i den diskrete modellen:

$$\begin{aligned}\Phi_{\beta}^a &= \Phi_{\beta}^g(t_{k+1}, t_k) = e^{F_a(t_{k+1}-t_k)} \\ &= e^{F_a \cdot \Delta t} = e^{0 \cdot \Delta t} = 1\end{aligned}\tag{3.26}$$

$$\begin{aligned}\tilde{Q}_{\beta}^a &= \int_{t_k}^{t_{k+1}} \Phi_{\beta}^a(t_{k+1}, \tau) G Q_{\beta}^a G^T \Phi_{\beta}^a(t_{k+1}, \tau) d\tau \\ &= \int_{t_k}^{t_{k+1}} Q_{\beta}^a d\tau = Q_{\beta}^a \Delta t\end{aligned}\tag{3.27}$$

Farget støy må også diskretiseres :

$$\Phi_{\gamma}^a = \Phi_{\gamma}^a(t_{k+1}, t_k) = e^{F_a \cdot \Delta t}\tag{3.28}$$

$$\begin{aligned}\tilde{Q}_{\gamma}^a &= \int_{t_k}^{t_{k+1}} \Phi_{\mu}^a(t_{k+1}, \tau) G Q_{\gamma}^a G^T \Phi_{\gamma}^a(t_{k+1}, \tau)^T d\tau \\ &= \int_{t_k}^{t_{k+1}} Q_{\gamma}^a d\tau = Q_{\gamma}^a \Delta t\end{aligned}\tag{3.29}$$

$$\begin{aligned}\Gamma_{\gamma}^a &= \Gamma_{\gamma}^a Q_{\gamma}^a (\Gamma_{\gamma}^a)^T = \int_{t_k}^{t_{k+1}} \Phi_{\gamma}^a(t_{k+1}, \tau) G Q_{\gamma}^a G^T \Phi_{\gamma}^a(t_{k+1}, \tau)^T d\tau \\ &= \Phi_{\gamma,12}^a(t_{k+1}, t_k) \Phi_{\gamma,22}^a(t_{k+1}, t_k)\end{aligned}\tag{3.30}$$



# Kapittel 4

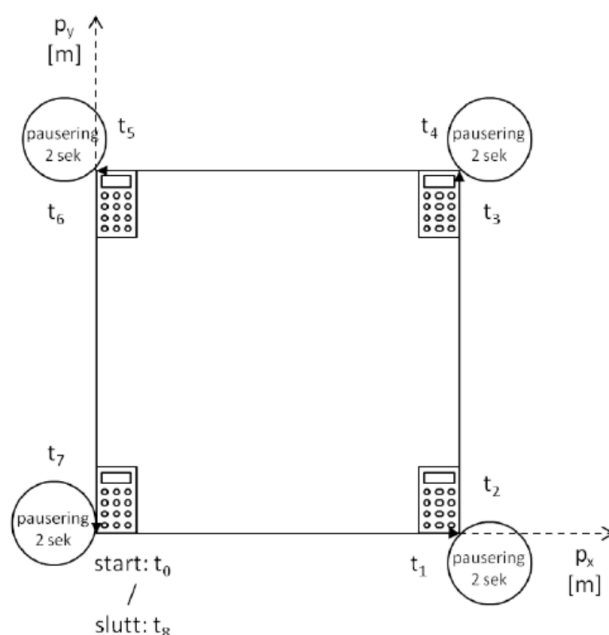
## Generering av gyro- og akselerometer målingene

### 4.1 Deterministisk banegenerator

Det er ønskelig å lage en deterministisk banegenerator i Matlab hvor hensikten er å motta verdier som kan testes opp mot en matematisk modell for et TNS. Interessante verdier som er nyttig å få ut av en banegenerator er legemets posisjon, hastighet og akselerasjon i forhold til tid. I tillegg til dette vil disse parameterne bli brukt til å regne ut spesifikke kraft og vinkelhastighet til legemet når legemet kjører gjennom banen.

#### 4.1.1 Legemets bane

Det simuleres ett omløp av en firkant av legemet i  $xy$ -planet. Banen består av rettlinjede bevegelser. I hvert hjørnet simuleres det at mobiltelefonen står i ro i 2 sekunder.



Figur 4.1: Banegenerator- omløp [5]

Likningene som beskriver bevegelsen av legemet langs en firkant i  $xy$ -planet. Handlingen starter i origo. Hastigheten er 0 i start- og sluttidspunktene.

Likningen for akselerasjon er:

$$a(t) = A \sin(\omega t) \quad (4.1)$$

Hastighetslikningen blir da:

$$\begin{aligned} v(t) &= v(0) + \int_0^t A \sin \omega t \, dt \\ &= -\frac{A}{\omega} (\cos \omega t - 1) \end{aligned} \quad (4.2)$$

Posisjonslikningen er :

$$\begin{aligned} p(t) &= p(0) + \int_0^t v(t) \, dt \\ &= -\frac{A}{\omega} \left[ \left( \frac{1}{\omega} \sin(\omega t) \right) - t \right] \end{aligned} \quad (4.3)$$

hvor  $A$  er amplitude,  $A = 0.3 \cdot \frac{\pi}{2}$  og  $\omega = \pi$ .

Legemet starter å bevege seg ved tiden  $t_0 = 0.00$ s og ender etter pause i siste hjørne ved tiden  $t_8 = 16.00$  s.

Generering av omløpet og pausering i hjørnene ser ut som følger

$t_0 - t_1$  er i ro i to sekunder,

$$\begin{aligned} \underline{a}''(t) &= \underline{0}; & \underline{p}''(t) &= \underline{0}; \\ \underline{v}''(t) &= \underline{0}; \end{aligned}$$

$t_0 - t_1$  0.3 m i  $p_x$  - retningen i løpet av to sekunder,

$$\begin{aligned} \underline{a}''(t) &= \begin{bmatrix} a(t) \\ 0 \\ 0 \end{bmatrix}; & \underline{p}''(t) &= \begin{bmatrix} p(t) \\ 0 \\ 0 \end{bmatrix}; \\ \underline{v}''(t) &= \begin{bmatrix} v(t) \\ 0 \\ 0 \end{bmatrix}; \end{aligned}$$

$t_1 - t_2$  er i ro i to sekunder,

$$\begin{aligned}\underline{a}^n(t) &= \underline{0}; & \underline{p}^n(t) &= \underline{0}; \\ \underline{v}^n(t) &= \underline{0};\end{aligned}$$

$t_2 - t_3$  0.3 m i  $p_y$  - retningen i løpet av to sekunder,

$$\begin{aligned}\underline{a}^n(t) &= \begin{bmatrix} 0 \\ a(t-t_2) \\ 0 \end{bmatrix}; & \underline{v}^n(t) &= \begin{bmatrix} 0 \\ v(t-t_2) \\ 0 \end{bmatrix}; \\ \underline{p}^n(t) &= p(t_2) + \begin{bmatrix} 0 \\ p(t-t_2) \\ 0 \end{bmatrix};\end{aligned}$$

$t_3 - t_4$  er i ro i to sekunder,

$$\begin{aligned}\underline{a}^n(t) &= \underline{0}; & \underline{p}^n(t) &= \underline{0}; \\ \underline{v}^n(t) &= \underline{0};\end{aligned}$$

$t_4 - t_5$  0.3 m i negativ  $p_x$  - retningen i løpet av to sekunder,

$$\begin{aligned}\underline{a}^n(t) &= \begin{bmatrix} -a(t) \\ 0 \\ 0 \end{bmatrix}; & \underline{p}^n(t) &= p(t_4) + \begin{bmatrix} -p(t-t_4) \\ 0 \\ 0 \end{bmatrix}; \\ \underline{v}^n(t) &= \begin{bmatrix} -v(t) \\ 0 \\ 0 \end{bmatrix};\end{aligned}$$

$t_5 - t_6$  er i ro i to sekunder,

$$\begin{aligned}\underline{a}^n(t) &= \underline{0}; & \underline{p}^n(t) &= \underline{0}; \\ \underline{v}^n(t) &= \underline{0};\end{aligned}$$

$t_6 - t_7$  0.3 m i negativ  $p_y$  - retningen i løpet av to sekunder,

$$\underline{a}^n(t) = \begin{bmatrix} 0 \\ -a(t - t_6) \\ 0 \end{bmatrix}; \quad \underline{v}^n(t) = \begin{bmatrix} 0 \\ -v(t - t_6) \\ 0 \end{bmatrix};$$

$$\underline{p}^n(t) = p(t_6) + \begin{bmatrix} 0 \\ -p(t - t_6) \\ 0 \end{bmatrix};$$

$t_7 - t_8$  er i ro i 2 sekunder

$$\underline{a}^n(t) = \underline{0}; \quad \underline{p}^n(t) = \underline{0};$$

$$\underline{v}^n(t) = \underline{0};$$

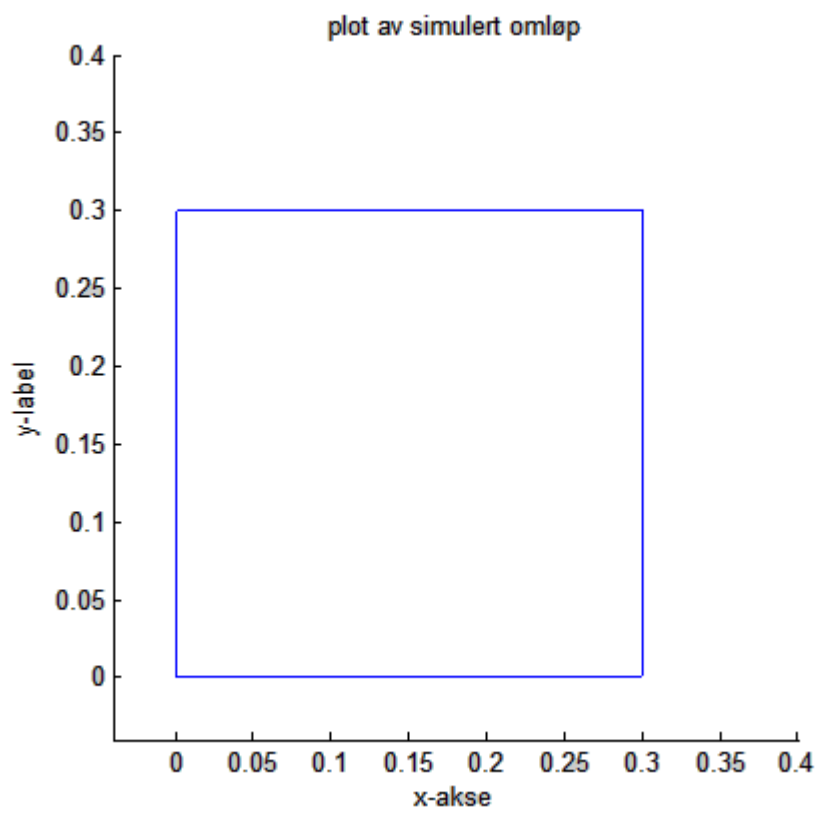
$$\underline{f}^n = \underline{a}^n + \underline{g}^n$$

$$\underline{f}^n = \begin{bmatrix} a(t) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$$

$$\underline{\omega}_a^{nn} = \underline{0}$$



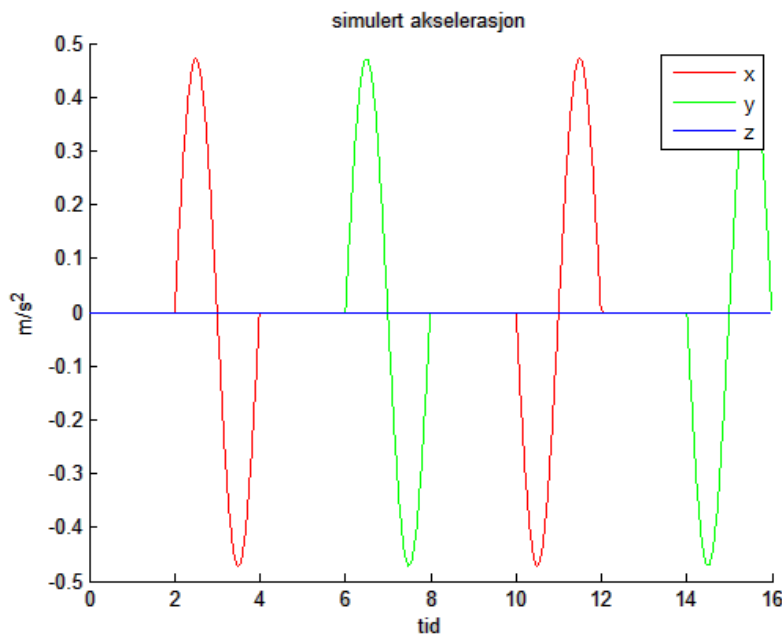
Figur 4.2 viser en banegenerator. Banegeneratoren genererer deterministiske data for bevegelse av legemet langs en bane i  $\{n\}$ -ramma.



Figur 4.2 viser banegenerator

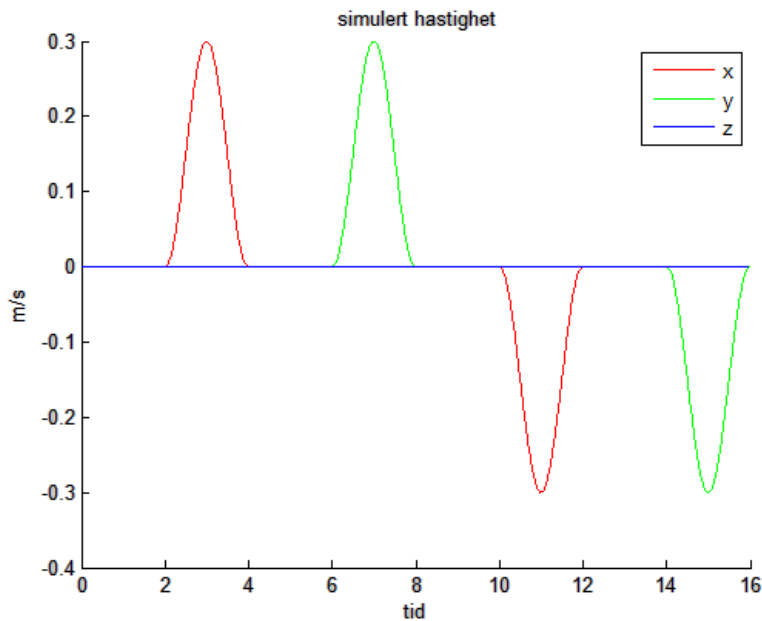
## 4.1.2 Simuleringsresultat fra banegenerator

Banegeneratoren gir ut simuleringsresultater for legemets tilstandsverdier i  $\{n\}$ -rammen, samt informasjonen for posisjon, hastighet og akselerasjon. Med disse resultatene er det bevist at spesifikk kraft og vinkelhastighet kan beregnes ut fra en deterministisk banegenerator. Krefte og akselerasjonene som vil påvirke systemet i denne sammenhengen er vist i figur 4.6, figur 4.7 og figur 4.8.



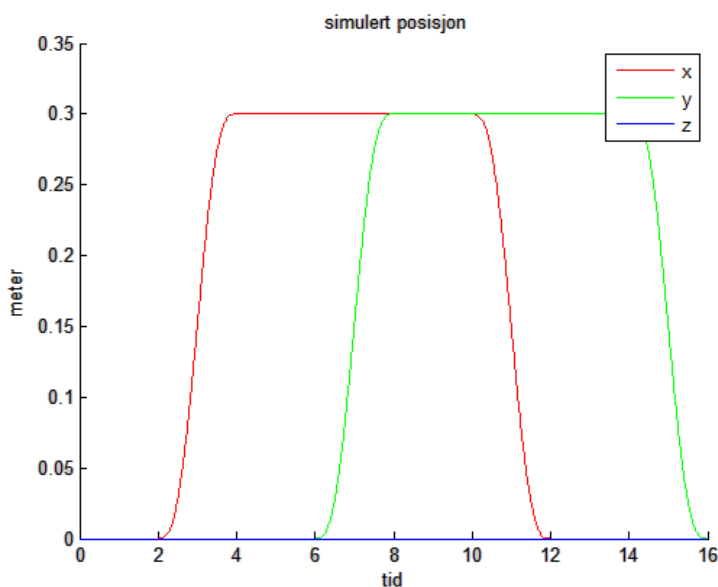
Figur 4.3 viser simuleringsresultat for akselerasjon

I figuren 4.3 ser vi at starten har akselerasjon 0 m/s<sup>2</sup> i 2 sekunder, etter 2 sekunder får x- retning en gradvis positiv påvirkning og reduseres gradvis ned. Det gjelder også y- retning. z- retning holdes til konstant null hele tiden. Dette stemmer veldig godt overens med det faktisk systemet ved å se på figuren 4.7



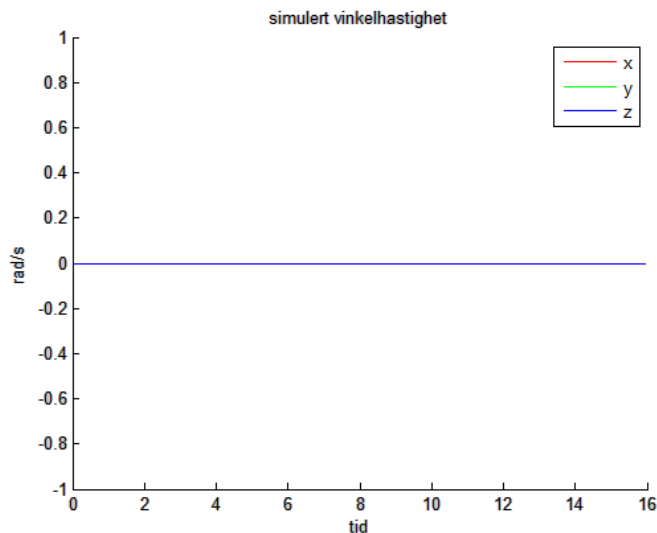
Figur 4.4 viser simuleringresultat for hastighet

Figur 4.4 viser hvordan hastigheten forandrer seg i x- og y- retning. Hastigheten er 0 m/s i start i 2 sekunder i x- retning. Etter det øker hastigheten til 0.3 m/s i samme retning, og minker til 0 i løpet av 2 sekunder. Følgende av dette gjør at hastigheten i x- retning minker etter 10 sekunder til -0.3 m/s og øker igjen. Det gjelder også y- retning, z-retning settes til null.



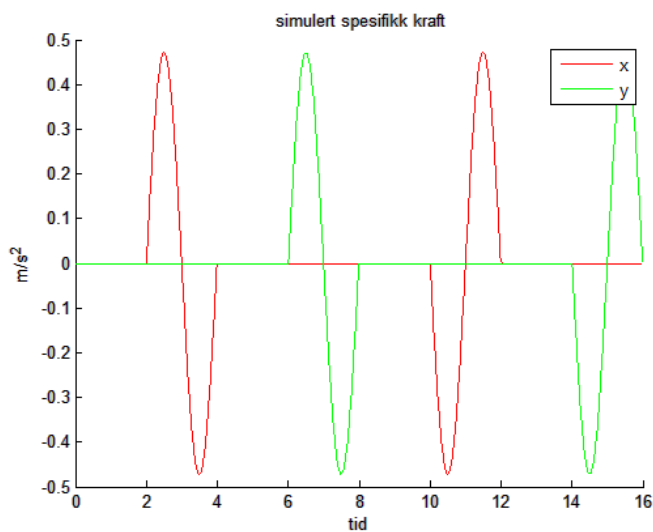
Figur 4.5 viser simuleringresultat for posisjon

Figur 4.5 viser hvordan posisjon forandrer seg i x-, y- og z- retninger. z- retning holdes null hele tiden. I x- retningen står posisjonen stille i 2 sekunder, og stiger deretter til 0.3 m, mens y- retning holdes konstant ved null. Det motsatte skjer når legemet skifter retning.

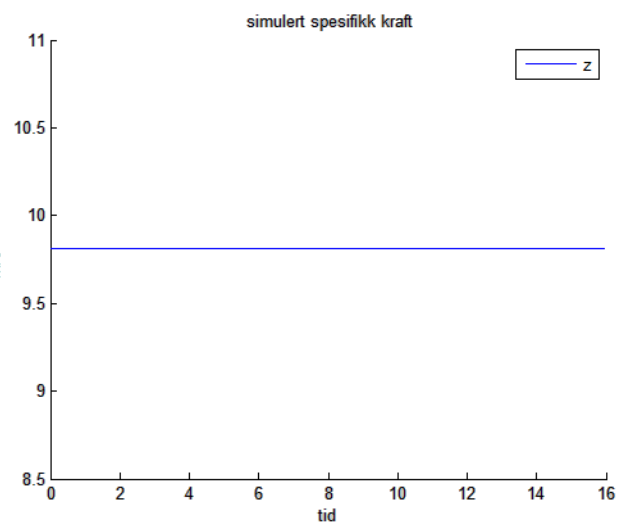


Figur 4.6 viser simuleringresultat for vinkelhastighet

Vinkelhastigheten holdes konstant på null hele tiden i x-, y- og z- komponentene ettersom tiden utvikler seg.



Figur 4.7 viser simulert spesifikk kraft i x- og y- retninger.



Figur 4.8 viser simulert spesifikk kraft i z- retning.

Ved å se på figur 4.3, figur 4.7 og figur 4.8 kan det konkluderes med at disse er tilnærmet identiske og stemmer veldig godt overens med resultat trukket ut fra legemets akselerasjon. I tillegg er gravitasjonskraften, som påvirker legemets tilstand i z-retning, med på å beskrive den spesifikke kraften. Dette kan sees fra figuren sammenliknet med figuren for akselerasjon.

## 4.2 Stokastiske gyro- og akselerometer-målinger

For en **trekset-akselerometer** er den generelle målelikningen definert som

$$\underline{\tilde{f}}^a = \underline{f}^a + \underline{\gamma}^a + \underline{\beta}^a + \underline{v}_f^a \quad (4.4)$$

Komponentene som inngår i likningen ser slik ut på vektorform

$$\underline{f}^a = [f_x^a; f_y^a; f_z^a] \quad (4.5)$$

$$\underline{\gamma}^a = [\gamma_x^a; \gamma_y^a; \gamma_z^a] \quad (4.6)$$

$$\underline{\beta}^a = [\beta_x^a; \beta_y^a; \beta_z^a] \quad (4.7)$$

For en **enakset-akselerometer** vil det se slik ut:

$$f = f_x^a \quad (4.8)$$

$$\tilde{f} = \tilde{f}_x^a \quad (4.9)$$

$$\gamma = \gamma_x^a \quad (4.10)$$

$$\beta = \beta_x^a \quad (4.11)$$

$\gamma^a$  og  $\beta^a$  beskrives av differensiallikningene

$$\dot{\gamma}^a = -\frac{1}{T_a} \gamma_a + v_\gamma \quad (4.12)$$

Dette er differensiallikningen til 1. ordens gauss- markov prosess som er definert tidligere i kapitlet 3 i likning 3. Likningen modellerer farget støy i akselerometer.

Den andre differensiallikningen er

$$\dot{\beta}^a = v_\beta^a \quad (4.13)$$

som beskriver støyprosessen browns bevegelse i akselerometer. De andre komponentene i 4.4 er  $v_f^a$  som representerer hvit støy- prosessen i akselerometer.

Utifra beskrivelsen over kan utlede en tilstandsrommodell for støyen, som har formen

$$\dot{\underline{x}}_a = \begin{bmatrix} \dot{\beta}^a & \dot{\gamma}^a \end{bmatrix}^T, \quad \underline{x}_a = \begin{bmatrix} \beta^a & \gamma^a \end{bmatrix}^T \quad (4.14)$$

Denne tilstandsromformen kan beskrives som et kontinuerlig system med tilhørende diskret målelikning.

$$\dot{\underline{x}}_a = F_a \underline{x}_a + G \underline{v}_a \quad (4.15)$$

$$\underline{z}_k = H \underline{x}_k + \underline{w}_k \quad (4.16)$$

som på matriseform kan skrives som

$$\begin{bmatrix} \dot{\beta}^a \\ \dot{\gamma}^a \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{T_a} \end{bmatrix} \begin{bmatrix} \beta^a \\ \gamma^a \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (4.17)$$

I denne oppgaven måling  $\underline{z}_k = \underline{0}$  fordi hastighet er like 0 når vi holder å måle i banegenerator.

$\dot{\underline{x}}^a = F^a \underline{x}^a + G \underline{v}$  diskretiseres på følgende måte:

$F_a$  blir 0 siden det ikke er noen tilbakekobling på tilstanden browns bevegelse og  $G=1$ . Innsatt dette gir følgende verdier i den diskrete modellen:

$$\begin{aligned} \Phi^a &= \Phi^a(t_{k+1}, t_k) = e^{F^a \cdot (t_{k+1} - t_k)} \\ &= e^{F^a \cdot \Delta t} = e^{0 \cdot \Delta t} = 1 \end{aligned} \quad (4.18)$$

$$\begin{aligned} \tilde{Q}_\beta^a &= \int_{t_k}^{t_{k+1}} \Phi_\beta^a(t_{k+1}, \tau) G Q_\beta^a G^T \Phi_\beta^a(t_{k+1}, \tau) d\tau \\ &= \int_{t_k}^{t_{k+1}} Q_\beta^a d\tau = Q_\beta^a \Delta t \end{aligned} \quad (4.19)$$

Farget støy må også diskretiseres, og få støyen over på den diskrete formen:

$$\Phi^a = \Phi_\gamma^a(t_{k+1}, t_k) = e^{F_a \cdot \Delta t} \quad (4.20)$$

$$\begin{aligned} \tilde{Q}_\gamma^a &= \int_{t_k}^{t_{k+1}} \Phi_\mu^a(t_{k+1}, \tau) G \tilde{Q}_\gamma^a G^T \Phi_\gamma^a(t_{k+1}, \tau)^T d\tau \\ &= \int_{t_k}^{t_{k+1}} Q_\gamma^a d\tau = Q_\gamma^a \Delta t \end{aligned} \quad (4.21)$$

$$\begin{aligned} \Gamma_\gamma^a &= \Gamma_\gamma^a Q_\gamma^a (\Gamma_\gamma^a)^T = \int_{t_k}^{t_{k+1}} \Phi_\gamma^a(t_{k+1}, \tau) G Q_\gamma^a G^T \Phi_\gamma^a(t_{k+1}, \tau)^T d\tau \\ &= \Phi_{\gamma,12}^a(t_{k+1}, t_k) \Phi_{\gamma,22}^a(t_{k+1}, t_k) \end{aligned} \quad (4.22)$$

$\Gamma^a$  og  $Q^a$  spaltes på følgende måte:

$$S = UDU^T \quad (4.23)$$

hvor  $U$  og  $D$  finnes ved hjelp av cholesky- faktorisering. Siden  $S \geq 0$ , vil  $D \geq 0$  og diagonal på formen

$$D^{\frac{1}{2}} = \begin{bmatrix} \sqrt{(d_{11})} & 0 \\ 0 & \sqrt{(d_{nn})} \end{bmatrix} \quad (4.24)$$

Videre lar man  $Q^a$ , og man kan da regne ut  $S$  og  $\Gamma^a$  på følgende måte:

$$S = UDU^T = UD^{\frac{1}{2}} * I * D^{\frac{1}{2}} U^T \quad (4.25)$$

$$\Gamma^a = UD^{\frac{1}{2}}$$

I matlab har vi brukt følgende funksjoner i diskretisering [17]

$$\Gamma^a = \text{kp2dpGa}(F^a, G, Q^a, T_a) \quad (4.26)$$

For en **treakset-gyrometer** er den generelle målelikningen definert [11] som

$$\tilde{\omega}_g^{ig} = \omega_g^{ig} + \underline{\gamma}^g + \underline{\beta}^g + \underline{v}^g \quad (4.27)$$

Komponentene som inngår i likningen ser slik ut på vektorform

$$\omega_g^{ig} = [\omega_{g,x}^{ig}; \omega_{g,y}^{ig}; \omega_{g,z}^{ig}] \quad (4.28)$$

For en **enakset-gyrometer**, med for eksempel x-akse vil dette se slik ut

$$\omega = \omega_{g,x}^{ig} \quad (4.29)$$

$$\tilde{\omega} = \tilde{\omega}_{g,x}^{ig} \quad (4.30)$$

$$\gamma = \gamma_x^g \quad (4.31)$$

$$\beta = \beta_x^g \quad (4.32)$$

Gyromålingene for x-aksen modelleres som

$$\tilde{\omega}_k = \omega_k + \gamma_k + \beta_k + v_{\omega} \quad (4.33)$$

Det er ønskelig å sette opp systemet på formen

$$\underline{\dot{x}}^g = F^g \underline{x}^g + G \underline{v} \quad (4.34)$$

med matrisene

$$F^g = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{T_g} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{T_g} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_g} \end{bmatrix} \quad (4.35)$$

og prosessstøymatrisa

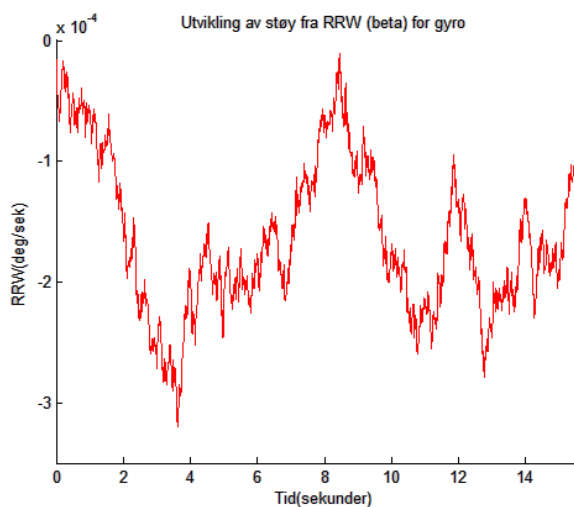
$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.36)$$



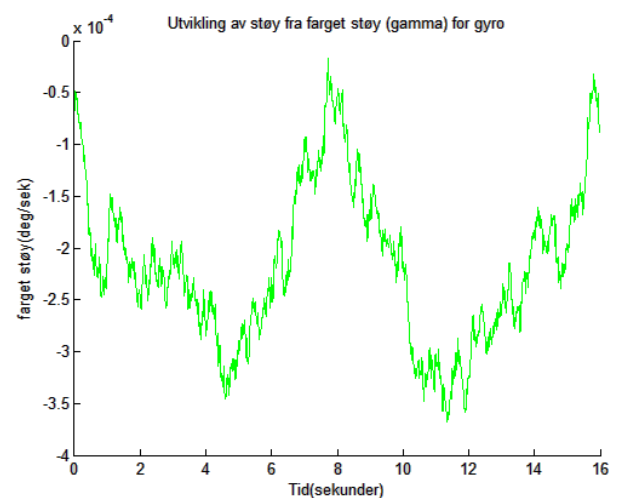
## 4.2.1 Simuleringsresultat for målefeil

Først simulerer vi støymodeller hver for seg i 16 sekunder. Hvis vi ser på figurene 4.9, 4.10, 4.11 og 4.12 så viser disse hvordan støyen utvikler seg hver av dem. Samlet er det disse støykildene som driver vinkelhastigheten og spesifikk kraft. En kan her se at alle støymodeller gir omtrent samme støyete signal. Hvis vi ser på figurene 4.13, 4.14, 4.15, 4.16, 4.17 og 4.18 viser et legemet som ligger i ro. Utgangsverdien fra sensorene blir lik null. På sensorsignalet er det alltid støy.

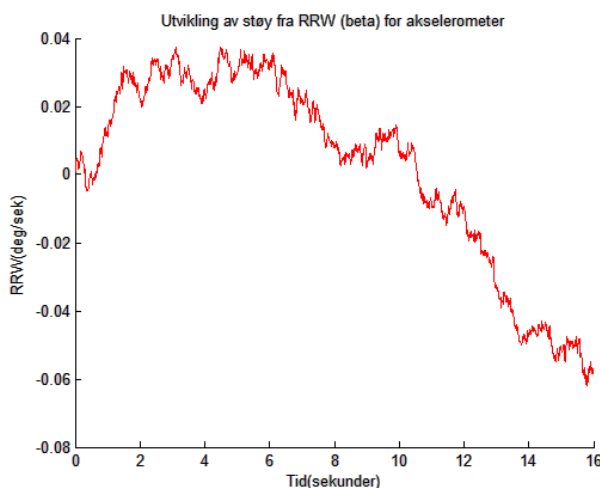
Under vises utskrift av utvikling av støyen simulert i 16 sekunder



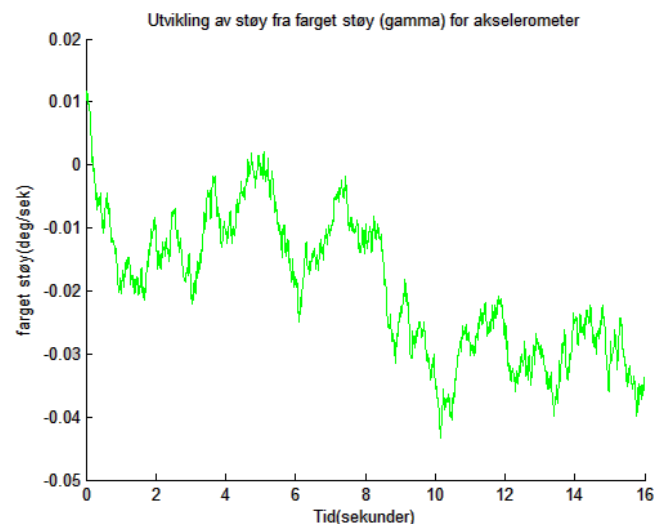
Figur 4.9 : Simulert utvikling av støybidrag fra browns bevegelse for gyro.



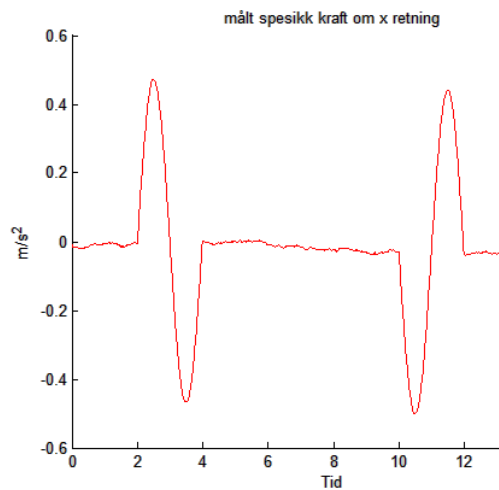
Figur 4.10: Simulert utvikling av støybidrag fra farget støy for gyro



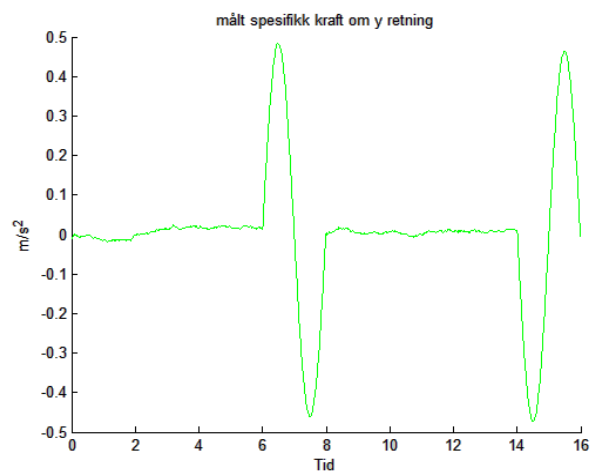
Figur 4.11: Simulert utvikling av browns bevegelse for akselerometer.



Figur 4.12: Simulert utvikling av farget støy for akselerometer.

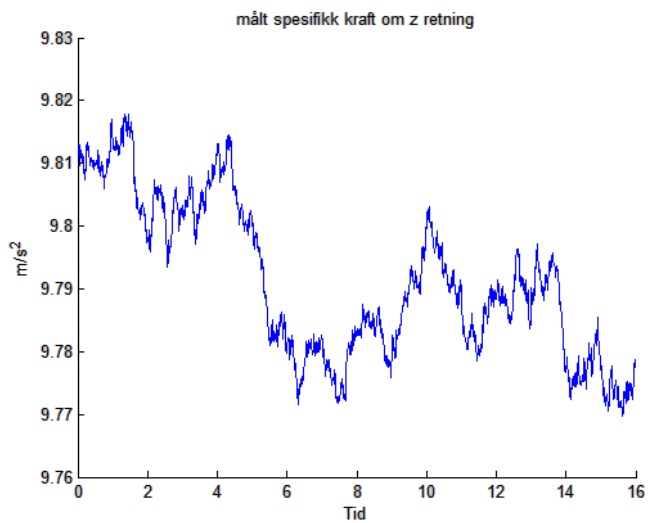


**Figur 4.13 : Målt spesifikk kraft om x- retning.**

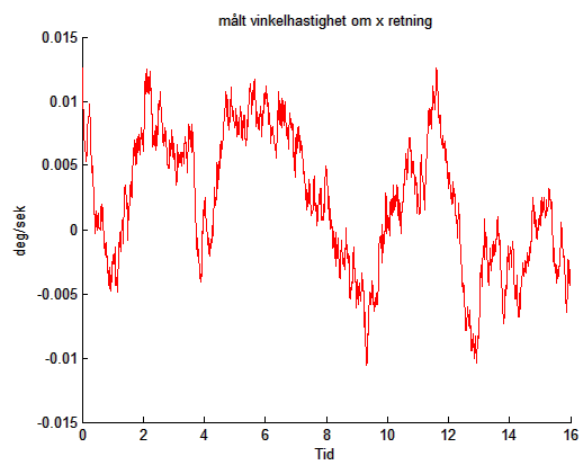


**Figur 4.14: Målt spesifikk kraft om y- retning.**

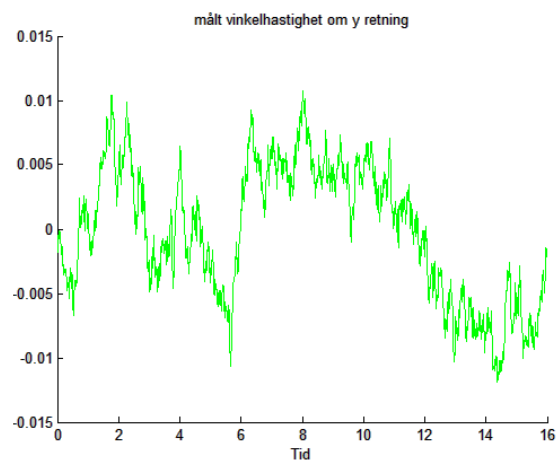
Figur 4.13 og 4.14 viser målt spesifikk kraft som har støy når legemet ligger i ro.



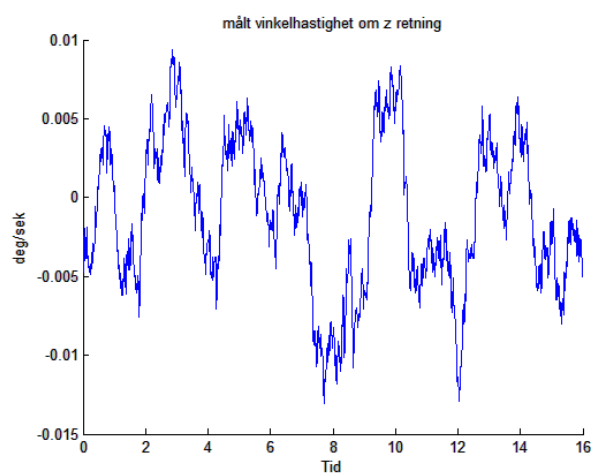
**Figur 4.15 : Målt spesifikk kraft om z- retning.**



Figur 4.16: Målt vinkelhastighet om x-retning.



Figur 4.17: Målt vinkelhastighet om y-retning



Figur 4.18: Målt vinkelhastighet om z-retning

## Pseudokode

% Initialisering

$$F^a = \text{diag} \left( \left[ 0, 0, 0, -\frac{1}{T^a}, -\frac{1}{T^a}, -\frac{1}{T^a} \right] \right)$$

$$F^g = \left( \left[ 0, 0, 0, -\frac{1}{T^g}, -\frac{1}{T^g}, -\frac{1}{T^g} \right] \right)$$

$$\mathbf{s}_\gamma^a = 10^{-3}$$

$$\bar{\mathbf{P}}_{\infty\gamma}^a = \left( \mathbf{s}_\gamma^a \right)^2$$

$$\bar{\mathbf{P}}_{o\gamma}^a = \bar{\mathbf{P}}_{\infty\gamma}^a$$

$$\tilde{\mathbf{Q}}_\gamma^a = \text{diag} \left( \left[ \frac{T^a}{2} \bar{\mathbf{P}}_{\infty\gamma}^a, \frac{T^a}{2} \bar{\mathbf{P}}_{\infty\gamma}^a, \frac{T^a}{2} \bar{\mathbf{P}}_{\infty\gamma}^a \right] \right)$$

$$\mathbf{s}_\gamma^g = 10^{-2}$$

$$\bar{\mathbf{P}}_{\infty\gamma}^g = \left( \mathbf{s}_\gamma^g \right)^2$$

$$\bar{\mathbf{P}}_{o\gamma}^g = \bar{\mathbf{P}}_{\infty\gamma}^g$$

$$\tilde{\mathbf{Q}}_\gamma^g = \text{diag} \left( \left[ \frac{T^g}{2} \bar{\mathbf{P}}_{\infty\gamma}^g, \frac{T^g}{2} \bar{\mathbf{P}}_{\infty\gamma}^g, \frac{T^g}{2} \bar{\mathbf{P}}_{\infty\gamma}^g \right] \right)$$

$$\mathbf{s}_\beta^a = 10^{-2}$$

$$\bar{\mathbf{P}}_{\infty\beta}^a = \left( \mathbf{s}_\beta^a \right)^2$$

$$\bar{\mathbf{P}}_{o\beta}^a = \bar{\mathbf{P}}_{\infty\beta}^a$$

$$\tilde{\mathbf{Q}}_\beta^a = \text{diag} \left( \left[ \frac{T^a}{2} \bar{\mathbf{P}}_{\infty\beta}^a, \frac{T^a}{2} \bar{\mathbf{P}}_{\infty\beta}^a, \frac{T^a}{2} \bar{\mathbf{P}}_{\infty\beta}^a \right] \right)$$

$$\mathbf{s}_\beta^g = 10^{-4}$$

$$\bar{\mathbf{P}}_{\infty\beta}^g = \left( \mathbf{s}_\beta^g \right)^2$$

$$\bar{\mathbf{P}}_{o\beta}^g = \bar{\mathbf{P}}_{\infty\beta}^g$$

$$\tilde{\mathbf{Q}}_\beta^g = \text{diag} \left( \left[ \frac{T^g}{2} \bar{\mathbf{P}}_{\infty\beta}^g, \frac{T^g}{2} \bar{\mathbf{P}}_{\infty\beta}^g, \frac{T^g}{2} \bar{\mathbf{P}}_{\infty\beta}^g \right] \right)$$

$$\dot{\underline{x}}^a = F^a \underline{x}^a + G \underline{v}^a$$

$$\dot{\underline{x}}^g = F^g \underline{x}^g + G \underline{v}^g$$

$$\underline{x}^a = [\underline{\beta}^a; \underline{\gamma}^a]$$

$$\underline{x}^g = [\underline{\beta}^g; \underline{\gamma}^g]$$

$$\underline{v}^a = [\underline{v}_\beta^a; \underline{v}_\gamma^a]$$

$$\underline{v}^g = [\underline{v}_\beta^g; \underline{v}_\gamma^g]$$

% Diskretisering

$$\Phi^a = e^{F^a dt};$$

$$\Phi^g = e^{F^g dt};$$

$$\tilde{Q}^g(1:3, 1:3) = \tilde{Q}_\beta^g;$$

$$\tilde{Q}^g(4:6, 4:6) = \tilde{Q}_\gamma^g;$$

$$\tilde{Q}^a(1:3, 1:3) = \tilde{Q}_\beta^a;$$

$$\tilde{Q}^a(4:6, 4:6) = \tilde{Q}_\gamma^a;$$

$$\Gamma^a = kp2dpga(F^a, G, \tilde{Q}^a, dt);$$

$$\Gamma^g = kp2dpga(F^g, G, \tilde{Q}^g, dt);$$

% Simulering av gyrometer og akselerometer

$$k = 1:N$$

$$\underline{x}_{k+1}^a = \Phi^a \underline{x}_k^a + \Gamma^a \underline{v}_k$$

$$\underline{x}_{k+1}^g = \Phi^g \underline{x}_k^g + \Gamma^g \underline{v}_k$$

% Beregning av målt spesifikk kraft og vinkelhastighet

$$\tilde{f}^b = f^b - (\underline{\beta}^b + \underline{\gamma}^b + \underline{v}_f^b);$$

$$\tilde{\omega}_b^{ib} = \omega_b^{ib} - (\underline{\beta}^g + \underline{\gamma}^g + \underline{v}_\omega^g);$$



# Kapittel 5

## TREGHETSNAVIGASJONS-SYSTEM

### 5.1 Tre-akset plattform med flat ikke-roterende jord

Oppgaven er begrenset til å navigere på en flat ikke –roterende jord i tre dimensjoner [14].

Transformasjonsfeilen er referert til n-systemet.

Feildefinisjoner		
$\delta \underline{\tilde{f}} = \underline{f}^b - \underline{\tilde{f}}^b$	$\delta \underline{x} = \underline{x}^n - \underline{\tilde{x}}^n$	$\underline{R}_b^n = R(\underline{\varepsilon}) \underline{\tilde{R}}_b^n$
$\delta \underline{\omega} = \underline{\omega}_{nb}^b - \underline{\tilde{\omega}}_{nb}^b$	$\delta \underline{v} = \underline{v}^n - \underline{\tilde{v}}^n$	$R(\underline{\varepsilon}) = I + S(\underline{\varepsilon})$

Tabell 5.1: Feildefinisjoner til tre-akset plattform

Fysisk system	Mekanisert system	Feillikninger
$\dot{\underline{p}}^n = \underline{v}^n$	$\dot{\underline{\tilde{p}}}^n = \underline{\tilde{v}}^n$	$\delta \dot{\underline{x}} = \delta \underline{v}^n$
$\dot{\underline{v}}^n = \underline{R}_b^n \underline{f}^b - \underline{g}^n$	$\dot{\underline{\tilde{v}}}^n = \underline{\tilde{R}}_b^n \underline{\tilde{f}}^b - \underline{g}^n$	$\delta \dot{\underline{v}}^n = S(\underline{\varepsilon}) \underline{\tilde{R}}_b^n \underline{\tilde{f}}^b - \underline{\tilde{R}}_b^n \delta \underline{f}$
$\dot{\underline{R}}_b^n = \underline{R}_b^n S(\underline{\omega}_{nb}^b)$	$\dot{\underline{\tilde{R}}}_b^n = \underline{\tilde{R}}_b^n S(\underline{\tilde{\omega}}_{nb}^b)$	$S(\underline{\dot{\varepsilon}}) = \underline{\tilde{R}}_b^n (\delta \underline{\tilde{\omega}}) \underline{\tilde{R}}_b^n \rightarrow \dot{\underline{\varepsilon}} = \underline{\tilde{R}}_b^n \delta \underline{\omega}_{nb}^b$

Tabell 5.2: Tre-akset plattform med ikke roterende jord

#### 5.1.1 Sann ulineær systemmodell

Denne modellen beskriver en sann deterministisk modell av det fysiske systemet. Systemet er gitt ved følgende likninger:

$$\dot{\underline{p}}^n = \underline{v}^n \quad (5.1)$$

$$\dot{\underline{v}}^n = \underline{R}_b^n \underline{f}^b - \underline{g}^n \quad (5.2)$$

$$\dot{\underline{R}}_b^n = \underline{R}_b^n S(\underline{\omega}_{nb}^b) \quad (5.3)$$

Der  $\underline{\dot{p}}$  står for hastighet,  $\underline{\dot{v}}$  står for akselerasjon og  $\dot{R}_b^n$  for orienteringen til legemet. Hastigheten og orienteringen gis ved gyromålinger,  $\underline{\omega}_{nb}^b$  og akselerometermålinger,  $\underline{f}^b$ .  $\underline{\omega}_{nb}^b$  og  $\underline{f}^b$  er gitt i koordinatsystemet til plattformen,  $\{b\}$ .

**Tilstander** De sanne verdiene for parametrene  $f, \omega, R$  og  $p$  i likningene for det fysiske systemet ((5.1)–(5.3)) er ukjente. Den sanne verdien av  $f$  og  $\omega$  ser slik:

$$\underline{f}^b = \tilde{\underline{f}}^b - \delta \underline{f}^b \quad (5.4)$$

$$\underline{\omega}_{nb}^b = \tilde{\underline{\omega}}_{nb}^b - \delta \underline{\omega}_{nb}^b \quad (5.5)$$

Disse to feildefinisjonene gir fremgang av likning (5.1)–(5.3):

$$\underline{\dot{p}}^n = \underline{v}^n \quad (5.6)$$

$$\underline{\dot{v}}^n = R_b^n (\tilde{\underline{f}}^b - \delta \underline{f}^b) - g^n \quad (5.7)$$

$$\dot{R}_b^n = R_b^n S(\tilde{\underline{\omega}}_{nb}^b - \delta \underline{\omega}_{nb}^b) \quad (5.8)$$

Tilstandsvektoren på standardformen blir en vektor med følgende tilstander:

$$\underline{x} = [\underline{p}^n \quad \underline{v}^n \quad R_b^n \quad \delta \underline{f}^b \quad \delta \underline{\omega}_{nb}^b]^T \quad (5.9)$$

Her representerer  $p$  posisjon og  $v$  hastighet i navigasjonsramme,  $\delta \underline{f}$  står for de tre feilkomponentene til akselerometeret,  $\delta \underline{\omega}$  for de tre feil-komponentene til gyroen.

**Pådrag** Det fysiske systemet har spesifikk kraft og vinkelhastighet som pådrag til navigasjonslikningene:

$$\underline{u}(t) = \begin{pmatrix} \tilde{\underline{f}}^b \\ \tilde{\underline{\omega}}_{nb}^b \end{pmatrix} = \begin{pmatrix} \underline{f}^b + \delta \underline{f}^b \\ \underline{\omega}_{nb}^b + \delta \underline{\omega}_{nb}^b \end{pmatrix} \quad (5.10)$$



**Prosesstøy** Prosesstøyen på akselerometerene og gyroen i IMU'en består av flere komponenter.

Tilstandsvektoren for feilkildene:

$$\underline{x}' = \begin{bmatrix} \underline{\gamma}^a \\ \underline{\gamma}^g \\ \underline{\beta}^a \\ \underline{\beta}^g \end{bmatrix} \quad (5.11)$$

Her  $\underline{\gamma}^a$  står for farget støy på akselerometerene,  $\underline{\gamma}^g$  står for farget støy på gyroene,  $\underline{\beta}^a$  står for bias feil i akselerometerene og  $\underline{\beta}^g$  står for bias feil i gyroene.

Den fargede støyen er en 1.ordens Markovprosess og biasen er konstant.

$$\dot{\underline{\gamma}}^a = \mathbf{T}_a \underline{\gamma}^a + \underline{v}_\gamma^a \quad (5.12)$$

$$\dot{\underline{\gamma}}^g = \mathbf{T}_g \underline{\gamma}^g + \underline{v}_\gamma^g \quad (5.13)$$

$$\dot{\underline{\beta}}^a = 0 \quad (5.14)$$

$$\dot{\underline{\beta}}^g = 0 \quad (5.15)$$

hvor tidskonstantene  $\mathbf{T}_a$  og  $\mathbf{T}_g$  er gitt ved:

$$\mathbf{T}_a = \begin{bmatrix} -\frac{1}{T_{a1}} & 0 & 0 \\ 0 & -\frac{1}{T_{a2}} & 0 \\ 0 & 0 & -\frac{1}{T_{a3}} \end{bmatrix} \quad (5.16)$$

$$\mathbf{T}_g = \begin{bmatrix} -\frac{1}{T_{g1}} & 0 & 0 \\ 0 & -\frac{1}{T_{g2}} & 0 \\ 0 & 0 & -\frac{1}{T_{g3}} \end{bmatrix} \quad (5.17)$$

Følgende lineær differensiallikning beskriver feilen, der  $\underline{v}'$  betegner hvit støy:

$$\dot{\underline{x}}' = F' \underline{x}' + G' \underline{v}' \quad (5.16)$$

Da kan man ved hjelp av denne feillikningen.

$$\delta \underline{f}^b = [\underline{\mu}^a + \underline{\beta}^a] + \underline{v}^a \quad (5.17)$$

$$\delta \underline{\omega}_{nb}^b = [\underline{\mu}^g + \underline{\beta}^g] + \underline{v}^g \quad (5.18)$$

**Målinger og målestøy** Målingene fra det fysiske systemet skjer i diskret og påvirkes av den diskrete hvite målestøyen,  $w_k$ :

$$\underline{z}_k = h(\underline{x}_k, w_k) \quad (5.19)$$

## 5.1.2 Navigasjonslikningene

Navigasjonslikningene [14] er også kalt en ulineære filtermodellen, beskrives med følgende likninger:

$$\dot{\underline{p}} = \underline{\tilde{v}}^n \quad (5.20)$$

$$\dot{\underline{\tilde{v}}} = \tilde{R}_b^n \underline{\tilde{f}}^b - \underline{g}^n \quad (5.21)$$

$$\dot{\tilde{R}}_b^n = \tilde{R}_b^n S(\underline{\tilde{\omega}}_{nb}^b) \quad (5.22)$$

**Tilstander** Dette systemet inneholder målte og predikerte verdier av posisjon, hastighet, vinkler, spesifikk kraft og vinkelhastighet.  $\underline{\tilde{f}}^b$  og  $\underline{\tilde{\omega}}_{nb}^b$  er kjente dersom man vet initialverdiene. De målte verdiene av posisjonen, hastighetene og orienteringen avviker fra de sanne verdiene på grunn av feil som oppstår under målingen. De målte verdiene defineres slik:

$$\delta \underline{p}^n = \underline{p}^n - \underline{\tilde{p}}^n \quad (5.23)$$

$$\delta \underline{v}^n = \underline{v}^n - \underline{\tilde{v}}^n \quad (5.24)$$

$$\underline{R}_b^n = (\mathbf{I} - S(\underline{\varepsilon})) \tilde{R}_b^n \quad (5.25)$$

**Pådrag** Pådraget genereres fra IMU'en. De målte IMU-kreftene og vinkelhastighetene får man ved å legge støyen til de sanne verdiene gitt av det fysiske systemet:

$$\underline{u}(t) = \begin{bmatrix} \tilde{f}^b \\ \tilde{\omega}_{nb}^b \end{bmatrix} \quad (5.26)$$

**Målinger og målestøy** Målingene skjer i diskret tid, og påvirkes ikke av målestøy.

$$\tilde{z}_k = h(\tilde{x}_k) \quad (5.27)$$

## 5.2 Integrasjonsrutine

I navigasjonslikninger for TNS benyttes to typer metoder. Først Euler's metode, etterfulgt av Heun's metode [21]. Begge metodene omgjør derivasjonsleddene til tilstandene og pådrag til form av  $(k+1)$  og integrasjonssekvensene blir enklere og predikere ved addisjonsregning.

Euler's metode baserer seg på nåværende punkt  $x_k$  til neste predikerte  $x_{k+1}$ , punktet evalueres og adderes med det tidligere punktet  $x_k$ .

Gitt den ordinære differensialligningen:

$$\dot{x} = f(t, x) \quad (5.28)$$

En tilnærming til Eulers metode er linearisering gjennom Taylorrekke-utvikling av funksjonen  $f$  med tidssteg  $T_s$

$$x(t_k) = \sum_{i=0}^{\infty} \frac{x^{(i)}(t_{k-1})}{i!} (t - t_{k-1})^i \quad (5.29)$$

$$x(t_k) = x(t_{k-1}) + T_s \dot{x}(t_{k-1}) + \frac{1}{2} T_s^2 \ddot{x}(t_{k-1}) + \dots \quad (5.30)$$

Euler's forovermetode inkluderer bare 1.ordens elementene og det gir :

$$x_{k+1} = x_k + T_s * f(t_k, x_k) \quad (5.31)$$

Med euler's forovermetode er tilnærmingen til den deriverte:

$$\dot{x}(k) \approx \frac{x(k+1) - x(k)}{T_s} \quad (5.32)$$

Euler's metode kan skrives som:

$$y_{k+1} = y_k + T_s f(t_k, y_k) \quad (5.33)$$

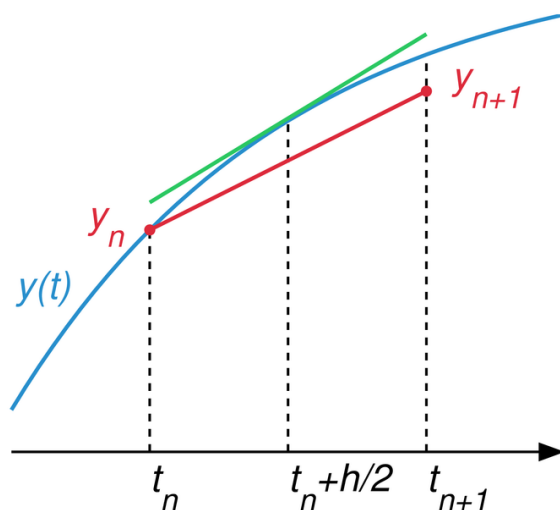
Heun's metode er 2.orden. Her beregner også nåværende og predikerte punktverdier, men i tillegg brukes de estimerte verdiene fra Euler's og legges til nåværende punkt av  $x_k$ .

Likningen for Heun's metode er gitt ved:

$$y'_{k+1} = y_k + T_s f(t_k, y_k) \quad (5.34)$$

$$y_{k+1} = y_k + \frac{T_s}{2} (f(t_k, y_k) + f(t_{k+1}, y'_{k+1})) \quad (5.35)$$

Med Heun's og Euler's metode vil navigasjonslikningene for det mekaniserte systemet få god nøyaktighet.



Figur 5.1: Diskretisering ved Heun's metode basert på Euler's metode [6].

## Pseudokode

### Euler's metode løsning av navigasjonslikninger

$$\underline{p}_{k+1} = \underline{p}_k + \underline{v}_k \cdot \Delta_t$$

$$\underline{v}_{k+1} = \underline{v}_k + \left( R_b^n \tilde{\underline{f}}_k - \underline{g}^n \right) \cdot \Delta_t$$

$$\tilde{R}_{k+1} = \tilde{R}_k + \tilde{R}_k S(\tilde{\omega}_k) \cdot \Delta_t$$

### Heun's metode løsning av navigasjonslikninger

$$\underline{p}'_{k+1} = \underline{p}_k + \underline{v}_k \cdot \Delta_t$$

$$\underline{v}'_{k+1} = \underline{v}_k + \left( R_b^n \tilde{\underline{f}}_k^b - \underline{g}^n \right) \cdot \Delta_t$$

$$\tilde{R}'_{k+1} = \tilde{R}_k + \tilde{R}_k S(\tilde{\omega}_k) \cdot \Delta_t$$

$$\underline{p}_{k+1} = \underline{p}_k + \frac{\Delta_t}{2} (\underline{v}_k + \underline{v}'_{k+1})$$

$$\underline{v}_{k+1} = \underline{v}_k + \frac{\Delta_t}{2} \left( \tilde{R}_k \tilde{\underline{f}}_k^b - \underline{g}^n + \tilde{R}'_{k+1} \tilde{\underline{f}}_{k+1}^b - \underline{g}^n \right)$$

$$\tilde{R}_{k+1} = \tilde{R}_k + \frac{\Delta_t}{2} \left( \tilde{R}_k S(\tilde{\omega}_k) + \tilde{R}'_{k+1} S(\tilde{\omega}_{k+1}) \right)$$

### % Feildefinisjoner

$$\delta \underline{p} = \underline{p}^n - \tilde{\underline{p}}^n$$

$$\delta \underline{v} = \underline{v}^n - \tilde{\underline{v}}^n$$

## 5.3 Oppsummering av TNS'et

Treghetsnavigasjonssystemet er oppsummet i figuren under:

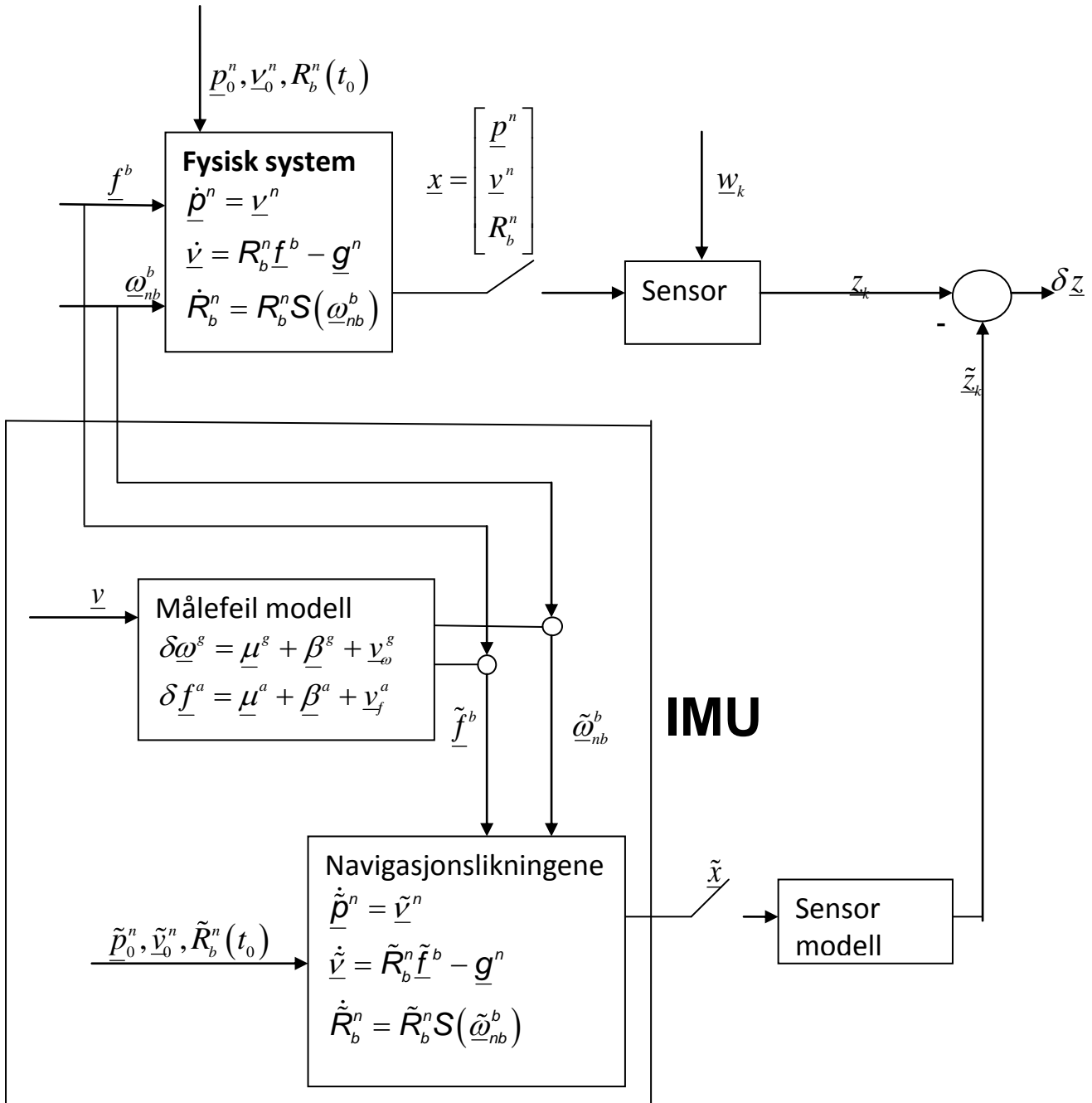
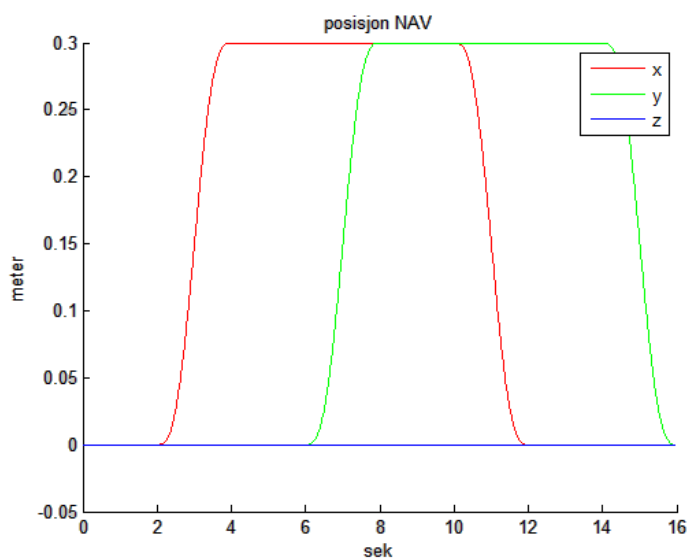


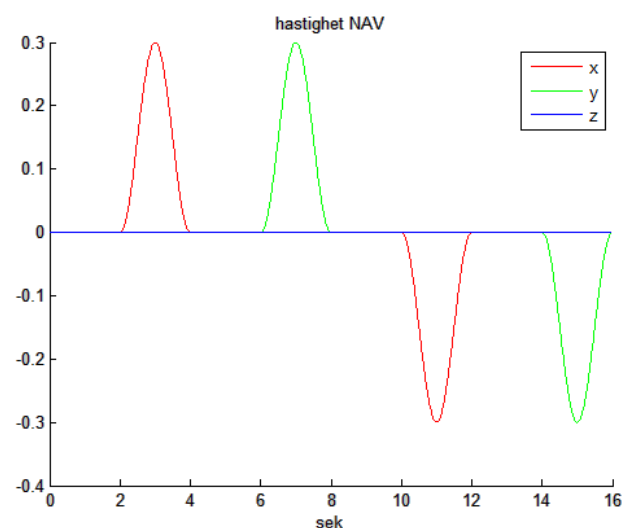
Figure 5.2: Oppsummering av Treghetsnavigasjonssystemet

## 5.4 Simuleringsresultat for TNS

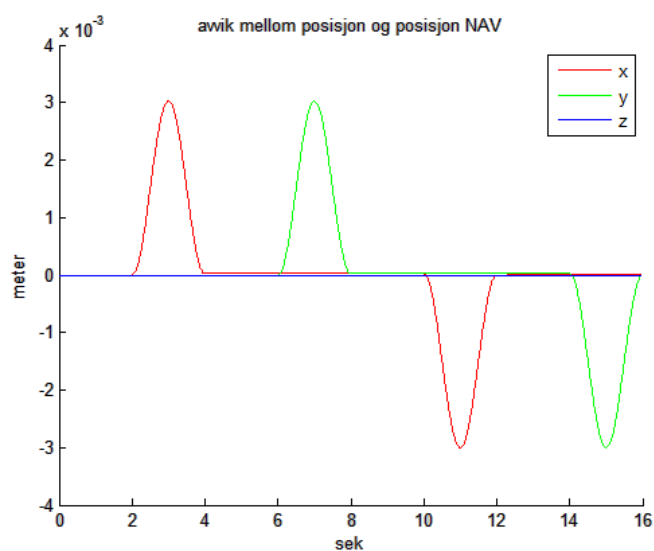
### 5.4.1 Simuleringsresultat for TNS ved bruk av Euler's metode ser slik ut:



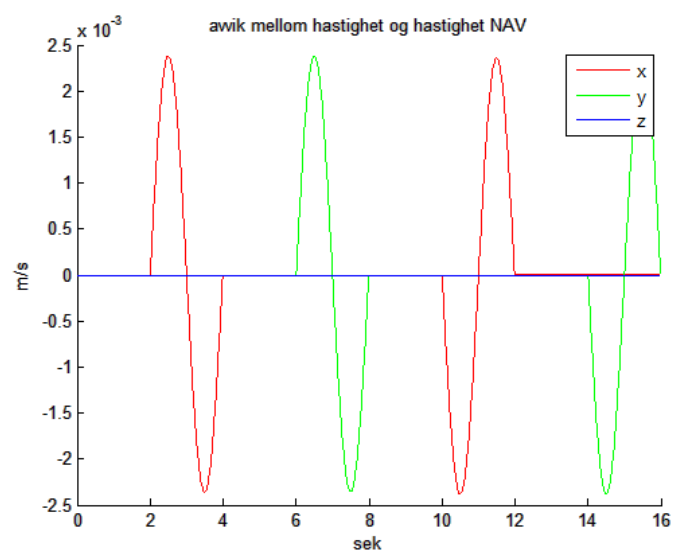
Figur 5.3 : Posisjon for navigasjon



Figur 5.4: Hastighet for navigasjon

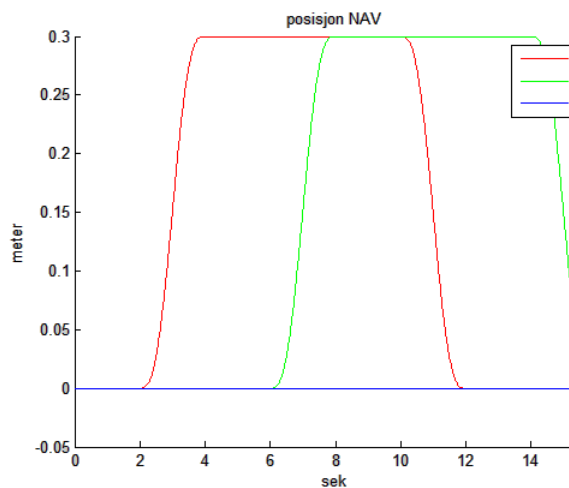


Figur 5.5 : Avvik mellom posisjon og posisjon NAV.  
Avvik ligger på  $\pm 3 \cdot 10^{-3}$  meter.

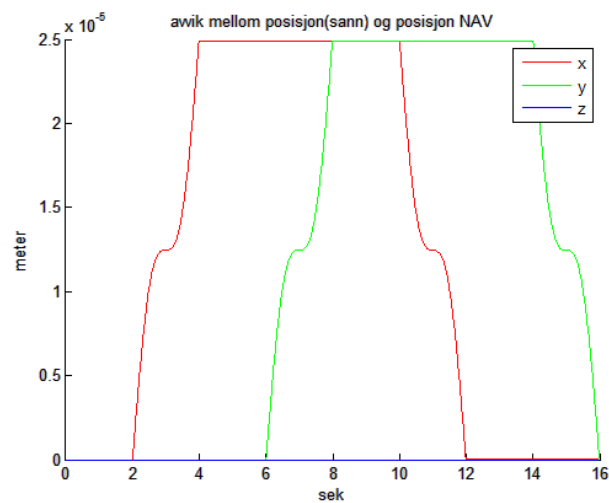


Figur 5.6: Avvik mellom simulert hastighet  
og hastighet for TNS.  
Avvik ligger på  $\pm 2.5 \cdot 10^{-3}$  m/s.

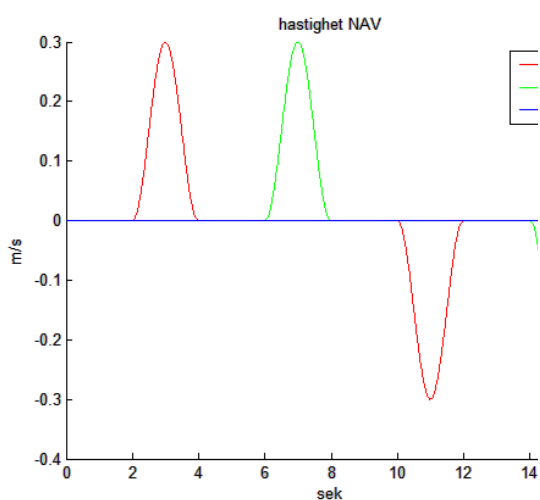
## 5.4.2 Simuleringsresultat ved bruk av Heun's metode ser slik ut:



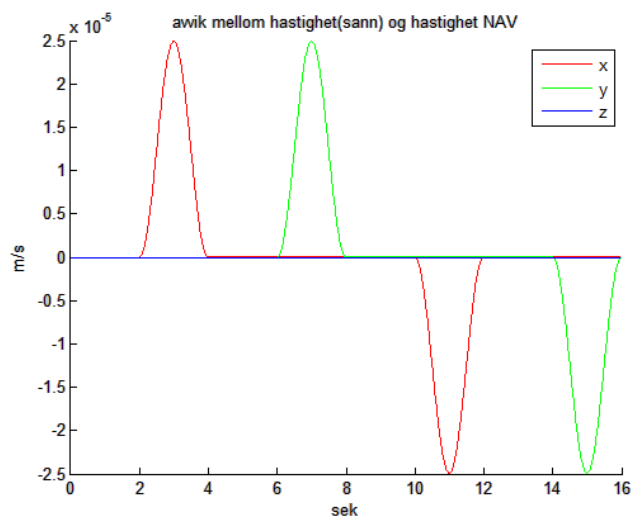
Figur 5.7 viser posisjon for NAV ved bruk av Heun's metode.



Figur 5.8 viser avvik mellom posisjon og posisjon for TNS. Avvik ligger på  $\pm 2.5 \cdot 10^{-5}$



Figur 5.9 viser hastighet for TNS



Figur 5.10 viser avvik mellom hastighet og Hastighet for TNS.

Avvik ligger på  $\pm 2.5 \cdot 10^{-5}$ .

De numeriske feilene som oppstår på grunn av integrasjonen, kunne vært modellert som feiltilstander. Av figur 5.8 og figur 5.10 ser vi at både er avviket svært mindre i forhold til sann verdi. Ved å se på figur 5.5 og 5.6 ser vi at både figurene ikke er avviket liten. Simuleringstiden er på 16 sekunder.



# Kapittel 6

## Kalmanfilter for TNS

Treghetsnavigasjonssystem er ikke en lineær prosess, men Kalmanfilteret kan tilpasses slik at det kan brukes i en ikke lineær prosess. I denne oppgaven vil brukt et linearisert Kalmanfilter som er beskrevet i kapittel 2.5.5.

### 6.1 Linearisert Kalmanfilter

Kalmanfiltre differensialligninger [18] kan skrives som:

$$\dot{\underline{x}} = F\underline{x} + G\underline{v} \quad (6.1)$$

Hvor  $F$  er systemmatrisa og  $G$  er prosesstøymatrisa.  $F$  er beskrevet av [15]

$$\delta \dot{\underline{p}}^n = \delta \underline{v}^n \quad (6.2)$$

$$\delta \dot{\underline{v}}^n = -S(\tilde{R}_b^n \tilde{f}^b) \underline{\varepsilon} + \tilde{R}_b^n \delta \underline{f} \quad (6.3)$$

$$\dot{\underline{\varepsilon}} = -S(\tilde{\omega}_b^{nb}) \underline{\varepsilon} + \tilde{R}_b^n \delta \underline{\omega}_b^{nb} \quad (6.4)$$

$$\dot{\underline{\gamma}}^a = T_a \underline{\gamma}^a + \underline{v}_\gamma^a \quad (6.5)$$

$$\dot{\underline{\gamma}}^s = T_s \underline{\gamma}^s + \underline{v}_\gamma^s \quad (6.6)$$

$$\dot{\underline{\beta}}^a = 0 \quad (6.7)$$

$$\dot{\underline{\beta}}^s = 0 \quad (6.8)$$

Tilstandsvektor til feilsystemet er gitt av differansen mellom de ovenfor gitte systemene, og beskrives som deltaverdier.

$$\delta \underline{x} = \begin{bmatrix} \delta \underline{p}^n & \delta \underline{v}^n & \underline{\varepsilon}^n & \underline{\gamma}^a & \underline{\gamma}^s & \underline{\beta}^a & \underline{\beta}^s \end{bmatrix}^T \quad (6.9)$$

Feilsystemet har ingen pådrag, det vil si  $\underline{u} = 0$ .

Den normalfordelte prosessstøyen som påvirker feilsystemet inneholder farget støy både bidrag fra både akselerometer og gyro:

$$\underline{v} = \begin{bmatrix} \underline{v}^a & \underline{v}^g & \underline{v}_\gamma^a & \underline{v}_\gamma^g & \underline{v}_\beta^a & \underline{v}_\beta^g \end{bmatrix} \quad (6.10)$$

Målingene som går inn i feilsystemet er diskrete, og definert til å være differansen mellom målinger fra den deterministiske banegeneratoren ( $\underline{z}_k$ ) og målinger fra navigasjonsligninger ( $\tilde{\underline{z}}_k$ ).

$$\delta \underline{z}_k = \underline{z}_k - \tilde{\underline{z}}_k \quad (6.11)$$

Feilsystemets systemmatrise  $F$ , og prosessstøymatrisen,  $G$ , blir til med bakgrunn i (6.2)–(6.8) og tilstandsvektoren  $\delta \underline{x}$  (6.9) slik:

$$\delta \dot{\underline{x}} = F(\delta \underline{x}) + G \underline{v} \quad (6.12)$$

### Feilsystem- oppsummering

$$\begin{bmatrix} \delta \dot{\underline{p}}^n \\ \delta \dot{\underline{v}}^n \\ \dot{\underline{\varepsilon}}^n \\ \dot{\underline{\gamma}}^a \\ \dot{\underline{\gamma}}^g \\ \dot{\underline{\beta}}^a \\ \dot{\underline{\beta}}^g \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -S(\tilde{\underline{f}}^b) & \tilde{\underline{R}}_b^n & 0 & \tilde{\underline{R}}_b^n & 0 \\ 0 & 0 & 0 & 0 & \tilde{\underline{R}}_b^n & 0 & \tilde{\underline{R}}_b^n \\ 0 & 0 & 0 & \underline{T}_a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \underline{T}_g & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \underline{p}^n \\ \delta \underline{v}^n \\ \underline{\varepsilon}^n \\ \underline{\gamma}^a \\ \underline{\gamma}^g \\ \underline{\beta}^a \\ \underline{\beta}^g \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \tilde{\underline{R}}_b^n & 0 & 0 & 0 & 0 & 0 \\ 0 & \tilde{\underline{R}}_b^n & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{v}^a \\ \underline{v}^g \\ \underline{v}_\gamma^a \\ \underline{v}_\gamma^g \\ \underline{v}_\beta^a \\ \underline{v}_\beta^g \end{bmatrix} \quad (6.13)$$

$$\delta \underline{z}_k = H(\underline{x}_k) + \underline{w}_k \quad (6.14)$$

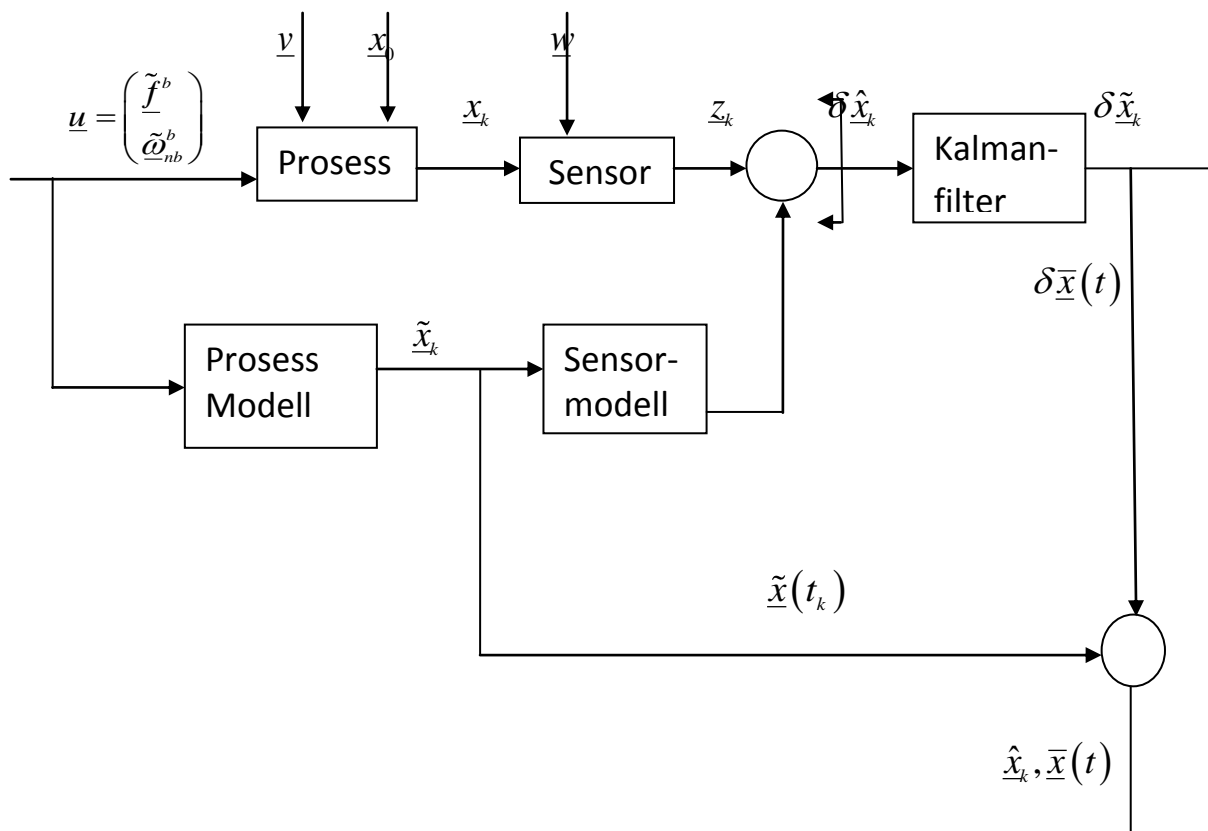


Figure 6.1: Blokkskjema av linearisert Kalmanfilter

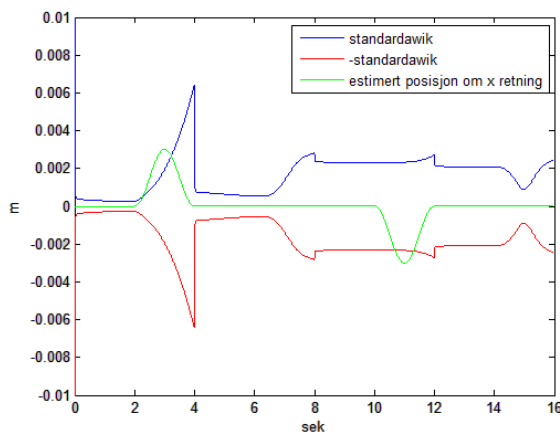
## 6.1.2 Simuleringsresultat for Kalmanfilteret

Ved analyse av kalmanfiltret plottes estimeringsfeilen til tilstandene med det tilhørende standardavviket. Standardavviket er definert som roten av diagonalelementene til kovariansmatrisen:

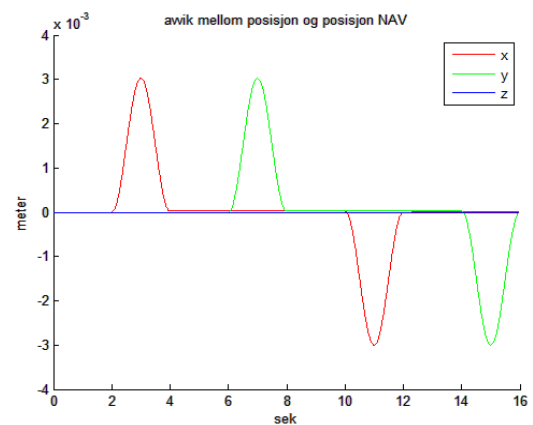
$$s = \text{sqrt}(\text{diag}(\hat{P})) \quad (6.15)$$

Feilestimatet er definert som:

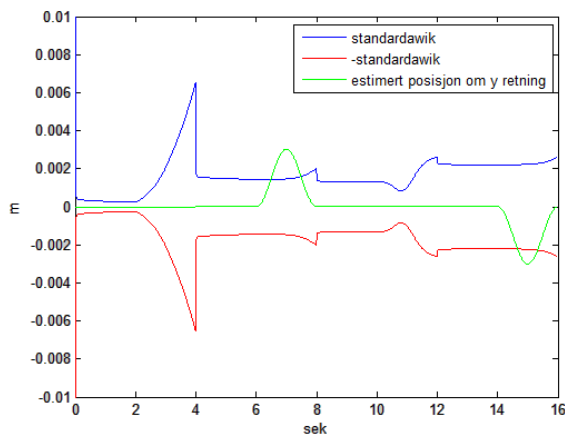
$$\delta \underline{x}_k = \underline{x}_k - \tilde{\underline{x}}_k \quad (6.16)$$



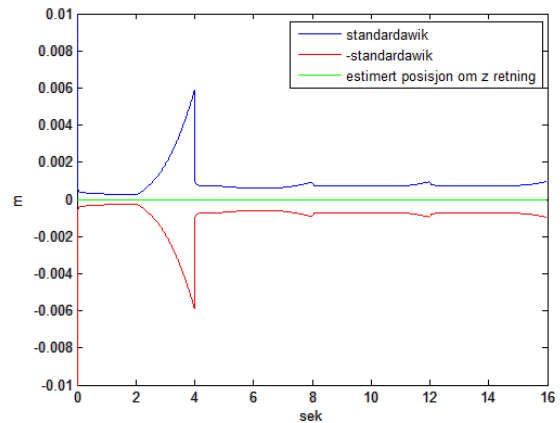
Figur 6.2: Estimert posisjon i x- retning med standardavvik.



Figur 5.5 : Avvik mellom posisjon og posisjon NAV.

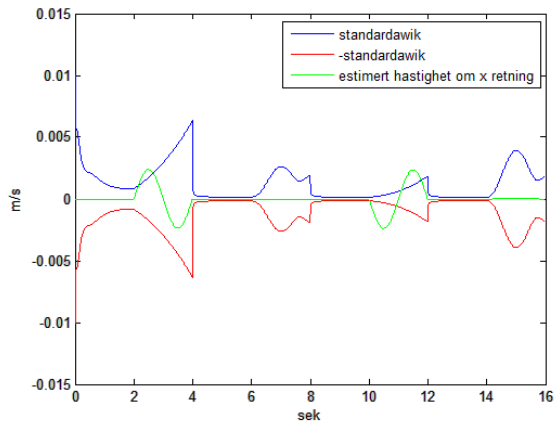


Figur 6.3 : Estimert posisjon i y- retning med standardavvik. Simulert i 16 sek.

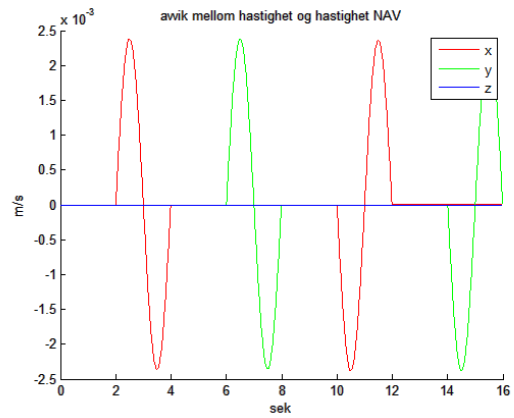


Figur 6.4 : Estimert posisjon i z- retning med standardavvik. Simulert i 16 sek.

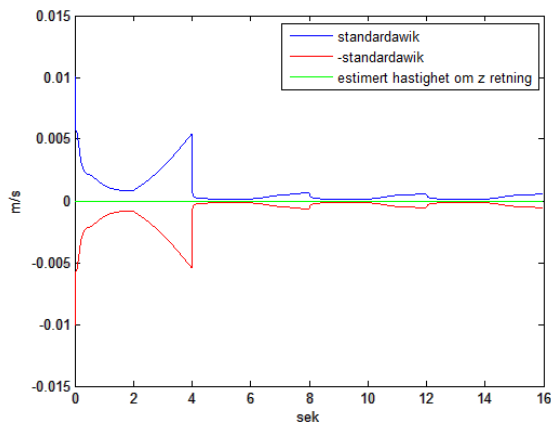
Ved å sammenlikne figur 5.5 med figurene 6.2, 6.3 og 6.4 ser man at et lineært kalmanfilter har klart å estimere den numeriske feilen i posisjon.



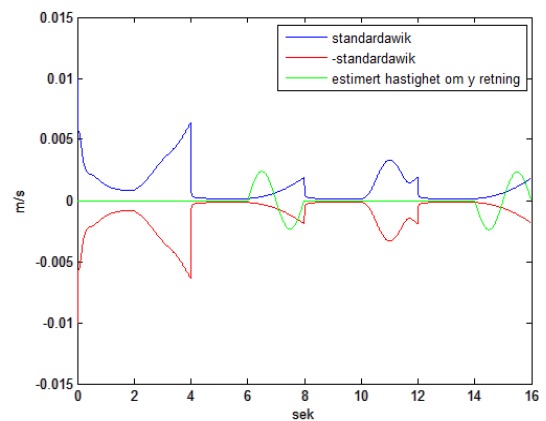
**Figur 6.5: Estimert hastighet i x- retning med standardavvik. Simulert i 16 sekunder.**



**Figur 5.6: Avvik mellom simulert hastighet og Hastighet NAV. Simulert i 16 sekunder.**

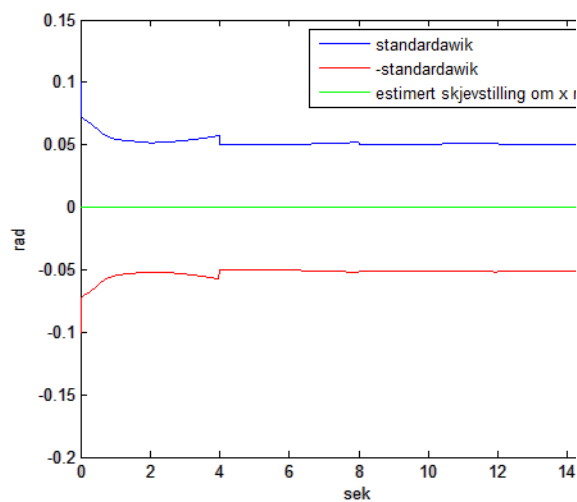


**Figur 6.6: Estimert hastighet i z- retning med standardavvik. Simulert i 16 sek.**

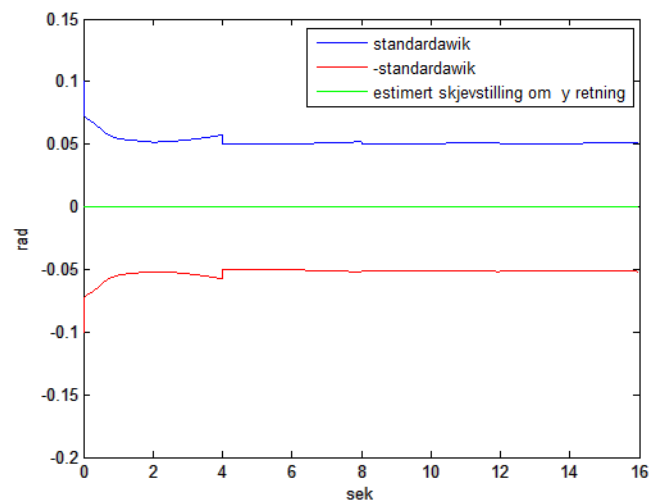


**Figur 6.7: Estimert hastighet i y- retning med standardavvik. Simulert i 16 sek**

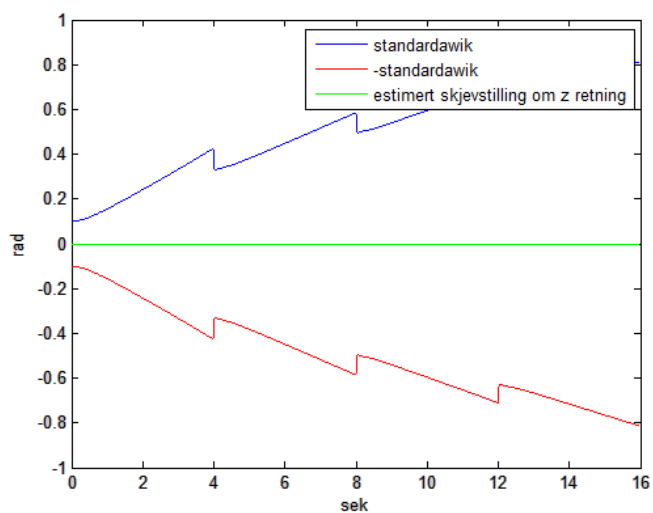
Ved å sammenlikne figur 5.6 med figurene 6.5, 6.6 og 6.7 ser vi at et lineært kalmanfilter har klart å estimere den numeriske feilen i hastighet.



Figur 6.8: Estimert skjevstilling i x- retning.



Figur 6.9: Estimert skjevstilling i y- retning.



Figur 6.10: Estimert skjevstilling i z- retning.

## Pseudokode for LKF

Initialiserings verdiene  $\tilde{\underline{x}}_0 = [\tilde{\underline{p}}^n, \tilde{\underline{v}}^n]$

Estimatet  $\delta\hat{\underline{x}}_0$  er satt til null og den predikerte kovarisansen og spektraltettheter til:

$$\hat{\mathbf{P}}_0 = \text{diag} \left( \left[ \underline{\sigma}_{pos}^2, \underline{\sigma}_{hast}^2, \underline{\sigma}_{skjevst}^2, \underline{\sigma}_{\gamma^a}^2, \underline{\sigma}_{\gamma^g}^2, \underline{\sigma}_{\beta^a}^2, \underline{\sigma}_{\beta^g}^2 \right] \right)$$

$$\tilde{\mathbf{Q}} = \text{diag} \left( \left[ \tilde{\mathbf{Q}}^a, \tilde{\mathbf{Q}}^g, \tilde{\mathbf{Q}}_{\gamma}^a, \tilde{\mathbf{Q}}_{\gamma}^g, \tilde{\mathbf{Q}}_{\beta}^a, \tilde{\mathbf{Q}}_{\beta}^g \right] \right)$$

$$\delta \underline{z}_k = -\tilde{\underline{p}}^n \quad \% \text{ posisjonsmåling}$$

$$\delta \underline{z}_k = -\tilde{\underline{v}}^n \quad \% \text{ hastighetsmåling}$$

% Tidsoppdatering

$$\mathbf{F} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -S(\underline{f}^b) & \tilde{\mathbf{R}}_b^n & 0 & \tilde{\mathbf{R}}_b^n & 0 \\ 0 & 0 & 0 & 0 & \tilde{\mathbf{R}}_b^n & 0 & \tilde{\mathbf{R}}_b^n \\ 0 & 0 & 0 & \mathbf{T}_a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{T}_g & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \tilde{\mathbf{R}}_b^n & 0 & 0 & 0 & 0 & 0 \\ 0 & \tilde{\mathbf{R}}_b^n & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[\Phi, \Gamma] = \text{k2dp}(F, G, Q, \Delta_t)$$

$$\delta \bar{\underline{x}}_k = \Phi \delta \bar{\underline{x}}_k$$

$$\bar{\underline{x}}_k = \tilde{\underline{x}}_k + \delta \bar{\underline{x}}_k$$

$$\bar{\mathbf{P}}_k = \Phi \hat{\mathbf{P}}_k \Phi^T + \Gamma Q \Gamma^T$$

% Måleoppdatering

$$\delta \hat{\underline{x}}_k = \delta \bar{\underline{x}}_k + \mathbf{K}_k (\delta \underline{z}_k - \mathbf{H}_k \delta \bar{\underline{x}}_k)$$

$$\hat{\underline{x}}_k = \tilde{\underline{x}}_k + \delta \hat{\underline{x}}_k$$

$$\mathbf{K}_k = \bar{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \bar{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_d)^{-1}$$

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \bar{\mathbf{P}}_k$$



# Kapittel 7 Konklusjon

I denne oppgaven har det blitt laget en banegenerator. Banen består av rettlinjede bevegelser. I hvert hjørne simuleres det at legemet står i ro i 2 sekunder. Bevegelsen starter ved tiden  $t_0 = 0$  sek og ender etter pausering i siste hjørne ved tiden  $t_8 = 16$  sek. Dette har vist å gi et presis resultat.

Utleddning av simuleringsmodell som genererer målinger av spesifikk kraft og vinkelhastighet, har resultert i forventede simuleringsresultater.

For å løse navigasjonslikningene er det brukt Euler's metode og Heun's metode. Etter å ha simulert begge metodene har vi funnet at Heun's metode gir bedre resultat i forhold til Euler's metode.

For at hele TNS skal bli komplett, har vi utledet bestemte feillikninger for systemet som beregner avviket mellom det fysiske systemet og navigasjonslikningene. Dette avviket brukes dermed videre i et lineært Kalmanfilter som filtrerer for predikerte feil og gir estimat mer stabilitet. Siden navigasjonslikningene ikke har gitt ut riktig verdier, er det mulig med å rette opp med TNS. Dette har vist at et lineært kalmanfilter estimerer riktig posisjon og hastighet, sammenliknet med det sanne fysiske systemet. Dette har også vist at skjevstillingen mangler målefeil fordi initialverdiene er null både i kalmanfilteret, navigasjonslikninger og banegeneratoren.



# Kapittel 8

## Videre arbeid

Videre arbeid som kan være aktuelt er :

- Legge til støy på initialsverdiene i banegeneratoren.
- Implementere et kalmanfilter for overnevnt støy.
- Vise skjevstillinger og estimert skjevstillinger.
- Undersøke med Heun's og Euler's metode når  $\underline{\tilde{f}}^b$  og  $\tilde{\omega}_{nb}^b$  inneholder støyen.
- Prosesstøyen blir korrelert siden Heun's metode blir brukt to ganger.



# Bibliografi

- [1] Anders Berge: MEMS – basert nordsøker UIO 20120528
- [2] Arthur Gelb. Applied optimal estimation (1974).
- [3] 1-MATLAB 2012b Primer.pdf
- [4] E.Kreyszig. Advanced Engineering Mathematics.Wiley
- [5] Hågen Kyllø: Treghetsnavigasjon med smarttelefon, Msc-avhandling UIO 20120611
- [6] [http://en.wikipedia.org/wiki/Midpoint\\_method](http://en.wikipedia.org/wiki/Midpoint_method), oktober 2012
- [7] <http://freedownloadb.com/PDF/SMARTPHONE-HARDWARE-SENSORS-31745063.HTML>, november 2012
- [8][https://www.memsnet.org/mems/what\\_is.html](https://www.memsnet.org/mems/what_is.html)
- [9]<http://no.wikipedia.org/wiki/Akselerometer>, november 2012
- [10]<http://no.wikipedia.org/wiki/Gyroskop,november> , november 2012
- [11] Jørn Skarbø Grahn: Estimering av MEMS- gyroparameterer, Msc-avhandling UiO 20110629
- [12] M. S. GREWAL and A. P. ANDREWS. Kalman Filtring, Theory and Using Matlab. Wiley.
- [13] Sami Hamra: Treghetsnavigasjon implementert i Android, Msc-avhandling UiO 20110627
- [14] O. Hallingstad. Eksempler på TNS- modeller. Forelesningen, 2004.
- [15] O. Hallingstad. Forelesningsnotater: Matematisk modellering av dynamiske systemer, notat UNIK 4540, 2011.
- [16]O. Hallingstad. Matematisk modellering av dynamiske systemer. Kompendie i faget UNIK 4540, UiO, Oktober 2008.
- [17] O. Hallingstad. Forelesningsnotater: Stokastiske systemer, UNIK 4500, 2010.
- [18] O. Hallingstad. Standardmodeller og kalmanfilterlikninger, notat 1 UNIK 4500, 24. august 2005.

- [19] Paal Alexander Nerholm: MEMS Inertial Navigation System, Msc-avhandling UiO 20111219
- [20] R. M. Rogers. Applied Mathematics in integrated Navigation systems. AIAA.
- [21] U. M. Ascher and L. R. Petzold. Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations. Siam
- [22] Silje Høstmælingen: Integrert Navigasjon åpen og lukket sløyfe, Msc-avhandling NTNU 20030617

# A MATLAB-kode

## A.1 BG.m

```
% Simulering av sensordata fra banegenerator
% Målefeil
% Simulering av navigasjonsligninger
% Kalmanfilter
clear all; clc; close all;
% Global startbetingelsene
% tidsintervaller
Ni=200; dti=2;
N=8*Ni; dt=dti/(Ni-1);
t=(0:8*Ni)*dt;
om=[0;0;0]; % Rotation speed

% Parametre
w=pi;
A=0.3*pi/2; % amplitude
g=9.81;

%-----deterministisk banegenerering-----
%-----
% Initialisering

a=zeros(1,Ni); % Initialisering akselerasjonsreferanse
v=zeros(1,Ni); % Initialisering hastighetsreferanse
p=zeros(1,Ni); % Initialisering posisjonsreferanse

a__n=zeros(3,N); % matrise for akselerasjon
v__n=zeros(3,N); % matrise for hastighet
p__n=zeros(3,N); % matrise for posisjon

f__n=zeros(3,N); % spesifikk kraft i n-ramma
w__b_nb=zeros(3,N); % vinkelhastighet i n-ramma
evm=zeros(3,N);

ev=[0;0;0]; %Initielle eulervinkler i BG
R__n_b=e2R(ev);

% Bane

p=- (A/w) * (sin(w*t(1:Ni)) / w - t(1:Ni));
v=- (A/w) * (cos(w*t(1:Ni)) - 1);
a=A*sin(w*t(1:Ni));
% 0-2s Står i ro i A
p__n(1,1:Ni)=0; v__n(1,1:Ni)=0; a__n(1,1:Ni)=0;
% 2-4s Flyttes fra A til B
p__n(1,Ni+1:2*Ni)=p; v__n(1,Ni+1:2*Ni)=v; a__n(1,Ni+1:2*Ni)=a;
% 4-6s Står i ro i B
p__n(1,2*Ni+1:3*Ni)=p__n(1,2*Ni);
% 6-8s Flyttes fra B til C
p__n(1,3*Ni+1:4*Ni)=p__n(1,2*Ni);
p__n(2,3*Ni+1:4*Ni)=p; v__n(2,3*Ni+1:4*Ni)=v; a__n(2,3*Ni+1:4*Ni)=a;
% 8-10s Står i ro i C
p__n(1,4*Ni+1:5*Ni)=p__n(1,4*Ni);
```

```

p__n(2,4*Ni+1:5*Ni)=p__n(2,4*Ni);
% 10-12s Flyttes fra C til D
p__n(1,5*Ni+1:6*Ni)=p__n(1,5*Ni)-p;
v__n(1,5*Ni+1:6*Ni)=-v; a__n(1,5*Ni+1:6*Ni)=-a;
p__n(2,5*Ni+1:6*Ni)=p__n(2,5*Ni);
% 12-14s Står i ro i D
p__n(1,6*Ni+1:7*Ni)=p__n(1,6*Ni);
p__n(2,6*Ni+1:7*Ni)=p__n(2,6*Ni);
% 14-16s Flyttes fra C til D
p__n(1,7*Ni+1:8*Ni)=p__n(1,7*Ni);
p__n(2,7*Ni+1:8*Ni)=p__n(2,7*Ni)-p; v__n(2,7*Ni+1:8*Ni)=-v;
a__n(2,7*Ni+1:8*Ni)=-a;
% Spesifikk kraft
f__n(1:2,1:8*Ni)=a__n(1:2,1:8*Ni); f__n(3,1:8*Ni)=g;
f__b=R__n_b*f__n;

%-----
% -----Målefeil-----
%-----

% x=Fx+Gv
% x består av farget støy og beta
% matriser for tilstandsrommodell
T=0.01 ;
Ts=0.04;
T__g=1; % Tidskonstant for gyrometer
T__a=2; % Tidskonstant for akselerometer
fb__b=f__b; %???Brukes i KF
wb__b_nb=w__b_nb; %???Brukes i KF, er tidsvariant
F__g=diag([0,0,0,-1/T__g,-1/T__g,-1/T__g]); % F matrise for gyrometer

F__a=diag([0,0,0,-1/T__a,-1/T__a,-1/T__a]); % F matrise for
akselerometer

s_gamma_a=10^-3;
Pue_gamma_a=s_gamma_a^2;
P0_gamma_a=Pue_gamma_a;

Qb_gamma_a=diag([(T__a/2)*Pue_gamma_a,(T__a/2)*Pue_gamma_a,(T__a/2)*Pue_gamma_a]);

s_beta_a=10^-2;
Pue_beta_a=s_beta_a^2;
P0_beta_a=Pue_beta_a;

Qb_beta_a=diag([(T__a/2)*Pue_beta_a,(T__a/2)*Pue_beta_a,(T__a/2)*Pue_beta_a]);

s_gamma_g=10^-2;
Pue_gamma_g=s_gamma_g^2;
P0_gamma_g=Pue_gamma_g;

Qb_gamma_g=diag([(T__g/2)*Pue_gamma_g,(T__g/2)*Pue_gamma_g,(T__g/2)*Pue_gamma_g]);

s_beta_g=10^-4;
Pue_beta_g=s_beta_g^2;
P0_beta_g=Pue_beta_g;

Qb_beta_g=diag([(T__g/2)*Pue_beta_g,(T__g/2)*Pue_beta_g,(T__g/2)*Pue_beta_g]);

```



```

G=eye(6);

Pe0=diag([1,1,1,0.1^2,0.1^2,0.1^2]); %kovariansematrise
H=[1 1 1 1 1 1]

X__g(1:3,1)=chol(P0_beta_g,'lower')*randn(3,1);
X__g(4:6,1)=chol(P0_gamma_g,'lower')*randn(3,1);

X__a(1:3,1)=chol(P0_beta_a,'lower')*randn(3,1);
X__a(4:6,1)=chol(P0_gamma_a,'lower')*randn(3,1);

% Diskretisering:

Fi__g=expm(F__g*dt)
Qb_g(1:3,1:3)=Qb_beta_g;
Qb_g(4:6,4:6)=Qb_gamma_g;
Ga__g=kp2dpga(F__g,G,Qb_g,dt);

Fi__a=expm(F__a*dt);
Qb_a(1:3,1:3)=Qb_beta_a;
Qb_a(4:6,4:6)=Qb_gamma_a;
Ga__a=kp2dpga(F__a,G,Qb_a,dt);

% Simulering av støymodell

for k=1:N-1

    X__g(:,k+1)=Fi__g*X__g(:,k)+Ga__g*randn(6,1);

    X__a(:,k+1)=Fi__a*X__a(:,k)+Ga__a*randn(6,1);

end

for k=1:N
    fb__b=f__b-(X__a(1:3,1:N)+X__a(4:6,1:N)+Qb_beta_a*randn(3,N)); % målt
    spesifikk kraft

    wbl__b_nb=w__b_nb-(X__g(1:3,1:N)+X__g(4:6,1:N)+Qb_beta_g*randn(3,N)); %
    målt vinkelhastighet

end

%-----Navigasjonssimulering-----
%-----
% g-kraft {f}-ramma
g__n=[0;0;g];
% navigasjonslikninger
pb__n=zeros(3,N); vb__n=zeros(3,N);
pbm__n=zeros(3,N); vbm__n=zeros(3,N);
evbm=zeros(3,N);

pb__n(:,1)=[0;0;0]; % Inietiell posisjon
vb__n(:,1)=[0;0;0]; % Initiell hastighet
evb=[0;0;0]; % Initielle eulervinkler
Rb__b_n=zeros(3,3,N);
Rb__b_n(:,:,1) = e2R(evb); % Initiell RKM

```

```

evbm(:,1)=evb; % Initiell eulervinkl
pbm__n(:,1)=[0;0;0];
vbm__n(:,1)=[0;0;0];

evbm=[0;0;0];
Rbm_b_n(:, :, 1)=e2R(evbm);

% Eulers metode

for k=1:N-1

    pb__n(:,k+1)=pb__n(:,k)+dt*v__b_n(:,k);
    vb__n(:,k+1)=vb__n(:,k)+dt*(Rb_b_n(:, :, k)*f__b(:,k)-g__n); %??? h.s. k-
1
    Rb_b_n(:, :, k+1)=Rb_b_n(:, :, k)+dt*(Rb_b_n(:, :, k)*S(wb_b_nb(:,k)));
    % Beregn eulervinkler
    evb(:,k+1)=R2e(Rb_b_n(:, :, k+1));
end

% Heuns metode

for k=1:N-1

    pbm__n(:,k+1)=pb__n(:,k)+dt*v__b_n(:,k);
    vbm__n(:,k+1)=vb__n(:,k)+dt*(Rb_b_n(:, :, k)*fb__b(:,k)-g__n);
    Rbm_b_n(:, :, k+1)=Rb_b_n(:, :, k)+dt*Rb_b_n(:, :, k)*S(wb_b_nb(:,k));

    pb__n(:,k+1)=pb__n(:,k)+(dt/2)*(vb__n(:,k)+vbm__n(:,k+1));
    vb__n(:,k+1)=vb__n(:,k)+(dt/2)*((Rb_b_n(:, :, k)*fb__b(:,k)-
g__n)+(Rbm_b_n(:, :, k+1)*fb__b(:,k+1)-g__n));

    Rb_b_n(:, :, k+1)=Rb_b_n(:, :, k)+dt*(Rb_b_n(:, :, k)*S(wb_b_nb(:,k))+Rbm_b_n(:, :,
k+1)*S(wb_b_nb(:,k+1)))/2;
    %3-2-1 Eulervinkler
    %evbm(:,k) =R2e(Rb_b_n);
    %evb(:,k)=R2e(Rb_b_n)+(dt/2)*R2e(Rb_b_n)');
end

%-----avvik mellom 'posisjon-p__n 'og 'posisjon(NAV)-p_b_n'---(euler's
metode)-----

dp__n=p__n-p_b_n;
dv__n=v__n-v_b_n;

%-----avvik mellom 'posisjon-p__n 'og 'posisjon(NAV)-p_b_n'---(heun's
metode)-----

dp1__n=p__n-pb__n;
dv1__n=v__n-vb__n;

```

```

%-----Kalmanfilter-----
%-----
%-----
I3=eye(3); c=[1 1 1]; g2r=pi/180; r2g=180/pi;

% Systemmatriser
% F og G har tidsinvariante undermatriser
F=zeros(21,21);
F(1:3,4:6)=I3;
F(10:12,10:12)=-1/T__a*I3;
F(13:15,13:15)=-1/T__g*I3;

G=zeros(21,18);
G(10:21,7:18)=eye(12);
% Initieell kovarians og spektralitettheter

sp_0_n=0.01; sv_0_n=0.01; sep_0_n=0.1*g2r; % [p]=m, [v]=m/s, [ep]=rad
sbe_0_a=0.001*g; sga_0_a=0.001*g; % [be_a]=m/s^2, [ga_a]=m/s^2
sbe_0_g=0.1*g2r; sga_0_g=0.1*g2r; % [be_g]=rad/s, [ga_g]=rad/s
% p v ep ga_a ga_g be_a be_g
Pe0=diag([sp_0_n*c,sv_0_n*c,sep_0_n*c,sga_0_a*c,sga_0_g*c,sbe_0_a*c,sbe_0_g*c].^2);
Pe=Pe0;

Qb=zeros(18,18);
qb__a=dt*(0.001*g)^2; qb_be_a=0.001^2; qb_ga_a=2*sga_0_a^2/T__a;
qb__g=dt*(0.1*g2r)^2; qb_be_g=(0.1*g2r)^2; qb_ga_g=2*sga_0_g^2/T__g;
Qb=diag([qb__a*c,qb__g*c,qb_ga_a*c,qb_ga_g*c,qb_be_a*c,qb_be_g*c])*10^-3;

dz1=-pb__n; H1=[I3 zeros(3,18)]; Rd1=0.001^2; % Posisjonsmåling
dz2=-vb__n; H2=[zeros(3,3) I3 zeros(3,15)]; Rd2=0.001^2; % Hastighetsmåling

%-----Kalmanfilter-----

dx0=zeros(21,1);dx=dx0;
xe0=zeros(6,N);
xe0(1:6,1)=[pb__n(:,1);vb__n(:,1)];
xe=xe0;
dxem=zeros(21,N); dxem(:,1)=dx0;

s=zeros(21,N);s(:,1)=sqrt(diag(Pe));

MOH1=[1:200];
MOH2=[401:600 801:1000 1201:1400];
for k=1:N-1
    % TO
    %???F og G oppdateres før diskretisering
    R=Rb_b_n(:, :, k);
    F(4:6,7:9)=-S(R*fb__b(:,k)); F(4:6,10:12)=R; F(4:6,16:18)=R;
    F(7:9,13:15)=R; F(7:9,19:21)=R;
    G(4:6,1:3)=R; G(7:9,4:6)=R;

```

```

[Fi,Ga]=k2dp(dt,F,G,Qb);
dxdp=Fi*dxe;
dxe=dxdp; %Lagres dersom vi bare har TO
k;
sqrt(Pe(1,1));
Pp=Fi*Pe*Fi'+Ga*Ga';
Pe=Pp; %Lagres dersom vi bare har TO
% MO
MO=0;
if ismember(k+1,MOH1)
    dz=dz1(:,k+1); H=H1; Rd=Rd1; MO=1;
elseif ismember(k+1,MOH2)
    dz=dz2(:,k+1); H=H2; Rd=Rd2; MO=1;
end
if MO
    K=Pp*H'*inv(H*Pp*H'+Rd); % diskret kalmanfilterforsterkning
    dxe=dxdp+K*(dz-H*dxdp);
    Pe=(eye(21)-K*H)*Pp; % estimerte kovarianse
end
dxem(:,k+1)=dxe;

s(:,k+1)=sqrt(diag(Pe));
end

%-----Plot banegenerator-----
figure()
clf
hold all,axis equal;
title('plot av simulert omløp')
axis([-0.04 0.40 -0.04 0.4])
xlabel('x-akse')
ylabel('y-label')
plot(p__n(1,:),p__n(2,:))

% Simulering av akselerasjon

figure()
clf
hold on;
title('simulert akselerasjon')
plot(0:0.01:15.99,a__n(1,1:8*Ni),'red');
plot(0:0.01:15.99,a__n(2,1:8*Ni),'green');
plot(0:0.01:15.99,a__n(3,1:8*Ni),'blue');
legend('x','y','z');
xlabel('tid')
ylabel('m/s^2')

figure()
clf
hold on;
title(' simulert hastighet')
plot(0:0.01:15.99,v__n(1,1:8*Ni),'red');
plot(0:0.01:15.99,v__n(2,1:8*Ni),'green');
plot(0:0.01:15.99,v__n(3,1:8*Ni),'blue');
legend('x','y','z');
xlabel('tid')
ylabel('m/s')

figure()
clf

```

```

hold on;
title(' simulert posisjon')
plot(0:0.01:15.99,p__n(1,1:8*Ni),'red');
plot(0:0.01:15.99,p__n(2,1:8*Ni),'green');
plot(0:0.01:15.99,p__n(3,1:8*Ni),'blue');
legend('x','y','z');
xlabel('tid')
ylabel('meter')

figure()
clf
hold on;
title('simulert vinkelhastighet')
plot(0:0.01:15.99,w__b_nb(1,1:8*Ni),'red');
plot(0:0.01:15.99,w__b_nb(2,1:8*Ni),'green');
plot(0:0.01:15.99,w__b_nb(3,1:8*Ni),'blue');
legend('x','y','z');
xlabel('tid')
ylabel('rad/s')

figure()
clf
hold on;
title(' simulert spesifikk kraft')
plot(0:0.01:15.99,f__n(1,1:8*Ni),'red');
plot(0:0.01:15.99,f__n(2,1:8*Ni),'green');
plot(0:0.01:15.99,f__n(3,1:8*Ni),'blue');
legend('x','y','z');
xlabel('tid')
ylabel('m/s^2')

%-----målefeil-----
%-----plot for beta(RRW) og my (farget støy)-----
figure()
hold on;
title('Utvikling av støy fra RRW (beta) for gyro')
plot(0:0.01:15.99,X__g(1,1:N),'red')
ylabel('RRW(deg/sek)')
xlabel('Tid(sekunder)')

figure()
hold on;
title('Utvikling av støy fra farget støy (gamma) for gyro')
plot(0:0.01:15.99,X__g(2,1:N),'green')
ylabel('farget støy(deg/sek)')
xlabel('Tid(sekunder)')

figure()
hold on;
title('Utvikling av støy fra RRW (beta) for akselerometer')
plot(0:0.01:15.99,X__a(1,1:N),'red')
ylabel('RRW(deg/sek)')
xlabel('Tid(sekunder)')

figure()
hold on;
title('Utvikling av støy fra farget støy (gamma) for akselerometer')
plot(0:0.01:15.99,X__a(2,1:N),'green')
ylabel('farget støy(deg/sek)')
xlabel('Tid(sekunder)')

```

```
%-----plot for målt spesifikk kraft og vinkelhastighet-----
----
```

```
figure()
hold on;
title ('målt spesifikk kraft om x retning')
plot(0:0.01:15.99,fb__b(1,1:N),'red')
xlabel('Tid')
ylabel('m/s^2')
```

```
figure()
hold on;
title('målt spesifikk kraft om y retning')
plot(0:0.01:15.99,fb__b(2,1:N),'green')
xlabel('Tid')
ylabel('m/s^2')
```

```
figure()
hold on;
title('målt spesifikk kraft om z retning')
plot(0:0.01:15.99,fb__b(3,1:N),'blue')
xlabel('Tid')
ylabel('m/s^2')
```

```
figure()
hold on
title('målt vinkelhastighet om x retning')
plot(0:0.01:15.99,wb1_b_nb(1,1:N),'red')
xlabel('Tid')
ylabel('deg/sek')
```

```
figure()
hold on
title('målt vinkelhastighet om y retning')
plot(0:0.01:15.99,wb1_b_nb(2,1:N),'green')
xlabel('Tid')
ylabel('deg/sek')
```

```
figure()
hold on
title('målt vinkelhastighet om z retning')
plot(0:0.01:15.99,wb1_b_nb(3,1:N),'blue')
xlabel('Tid')
ylabel('deg/sek')
```

```
%-----plot av NAV----- med Euler's metode
```

```
figure()
hold on;
title('posisjon NAV')
plot(0:0.01:15.99,p_b_n(1,1:8*Ni),'red');
plot(0:0.01:15.99,p_b_n(2,1:8*Ni),'green');
plot(0:0.01:15.99,p_b_n(3,1:8*Ni),'blue');
legend('x','y','z');
xlabel('sek')
ylabel('meter')
```

```
figure()
```

```

hold on;
title('hastighet NAV')
plot(0:0.01:15.99,v_b_n(1,1:8*Ni),'red');
plot(0:0.01:15.99,v_b_n(2,1:8*Ni),'green');
plot(0:0.01:15.99,v_b_n(3,1:8*Ni),'blue');
legend('x','y','z');
xlabel('sek')
ylabel('m/s')
%-----plot av avvik mellom posisjon p__n og p_b_n-----
figure()
hold on;
title('avvik mellom posisjon og posisjon NAV')
plot(0:0.01:15.99,dp__n(1,1:8*Ni),'red');
plot(0:0.01:15.99,dp__n(2,1:8*Ni),'green');
plot(0:0.01:15.99,dp__n(3,1:8*Ni),'blue');
legend('x','y','z');
xlabel('sek')
ylabel('meter')
%-----plot av avvik mellom hastighet v__n og vb__n-----
figure()
hold on;
title('avvik mellom hastighet og hastighet NAV')
plot(0:0.01:15.99,dv__n(1,1:8*Ni),'red');
plot(0:0.01:15.99,dv__n(2,1:8*Ni),'green');
plot(0:0.01:15.99,dv__n(3,1:8*Ni),'blue');
legend('x','y','z');
xlabel('sek')
ylabel('m/s')

%-----plot av NAV----- med heun's metode
figure()
hold on;
title('posisjon NAV')
plot(0:0.01:15.99,pb__n(1,1:8*Ni),'red');
plot(0:0.01:15.99,pb__n(2,1:8*Ni),'green');
plot(0:0.01:15.99,pb__n(3,1:8*Ni),'blue');
legend('x','y','z');
xlabel('sek')
ylabel('meter')

figure()
hold on;
title('hastighet NAV')
plot(0:0.01:15.99,vb__n(1,1:8*Ni),'red');
plot(0:0.01:15.99,vb__n(2,1:8*Ni),'green');
plot(0:0.01:15.99,vb__n(3,1:8*Ni),'blue');
legend('x','y','z');
xlabel('sek')
ylabel('m/s')

%-----plot av avvik mellom posisjon p__n og p_b_n-----
figure()
hold on;
title('avvik mellom posisjon(sann) og posisjon NAV')
plot(0:0.01:15.99,dp1__n(1,1:8*Ni),'red');
plot(0:0.01:15.99,dp1__n(2,1:8*Ni),'green');
plot(0:0.01:15.99,dp1__n(3,1:8*Ni),'blue');
legend('x','y','z');
xlabel('sek')
ylabel('meter')

```

```

%-----plot av avvik mellom hastighet v__n og vb__n-----
figure()
hold on;
title('avvik mellom hastighet(sann) og hastighet NAV')
plot(0:0.01:15.99,dv1__n(1,1:8*Ni),'red');
plot(0:0.01:15.99,dv1__n(2,1:8*Ni),'green');
plot(0:0.01:15.99,dv1__n(3,1:8*Ni),'blue');
legend('x','y','z');
xlabel('sek')
ylabel('m/s')

%-----plot for kalmanfilter-----
-----
figure();
plot(0:0.01:15.98,s(1,1:N-1),'b',0:0.01:15.98,-s(1,1:N-1),'r',0:0.01:15.98,(dp__n(1,1:N-1)-dxem(1,1:N-1)),'g');
legend('standardawik',' -standardawik','estimert posisjon om x retning')
xlabel('sek')
ylabel('m')
figure()
plot(0:0.01:15.98,s(2,1:N-1),'b',0:0.01:15.98,-s(2,1:N-1),'r',0:0.01:15.98,(dp__n(2,1:N-1)-dxem(2,1:N-1)),'g');
legend('standardawik',' -standardawik','estimert posisjon om y retning')
xlabel('sek')
ylabel('m')
figure()
plot(0:0.01:15.98,s(3,1:N-1),'b',0:0.01:15.98,-s(3,1:N-1),'r',0:0.01:15.98,(dp__n(3,1:N-1)-dxem(3,1:N-1)),'g');
legend('standardawik',' -standardawik','estimert posisjon om z retning')
xlabel('sek')
ylabel('m')

figure()
plot(0:0.01:15.98,s(4,1:N-1),'b',0:0.01:15.98,-s(4,1:N-1),'r',0:0.01:15.98,(dv__n(1,1:N-1)-dxem(4,1:N-1)),'g');
legend('standardawik',' -standardawik','estimert hastighet om x retning')
xlabel('sek')
ylabel('m/s')
figure()
plot(0:0.01:15.98,s(5,1:N-1),'b',0:0.01:15.98,-s(5,1:N-1),'r',0:0.01:15.98,(dv__n(2,1:N-1)-dxem(5,1:N-1)),'g');
legend('standardawik',' -standardawik','estimert hastighet om y retning')
xlabel('sek')
ylabel('m/s')
figure()
plot(0:0.01:15.98,s(6,1:N-1),'b',0:0.01:15.98,-s(6,1:N-1),'r',0:0.01:15.98,(dv__n(3,1:N-1)-dxem(6,1:N-1)),'g');
legend('standardawik',' -standardawik','estimert hastighet om z retning')
xlabel('sek')
ylabel('m/s')

figure()
plot(0:0.01:15.98,s(7,1:N-1)*r2g,'b',0:0.01:15.98,-s(7,1:N-1)*r2g,'r',0:0.01:15.98,(dxem(7,1:N-1)*r2g),'g');
legend('standardawik',' -standardawik','estimert skjevstilling om x retning')
xlabel('sek')

```



```

ylabel('rad')
figure()
plot(0:0.01:15.98,s(8,1:N-1)*r2g,'b',0:0.01:15.98,-s(8,1:N-
1)*r2g,'r',0:0.01:15.98,(dxem(8,1:N-1)*r2g),'g');
legend('standardawik','-standardawik','estimert skjevstilling om y
retning')
xlabel('sek')
ylabel('rad')
figure()
plot(0:0.01:15.98,s(9,1:N-1)*r2g,'b',0:0.01:15.98,-s(9,1:N-
1)*r2g,'r',0:0.01:15.98,(dxem(9,1:N-1)*r2g),'g');
legend('standardawik','-standardawik','estimert skjevstilling om z
retning')
xlabel('sek')
ylabel('rad')

```

## A.2 R\_b\_n.m

```

function R_b_n = y(a3,b2,c1)
% RKM for det kinematiske problem 3-2-1 Eulervinkler
a3=a3*pi/180;      b2=b2*pi/180;      c1=c1*pi/180;
R_b_n =[cos(a3)*cos(b2) cos(a3)*sin(b2)*sin(c1)-sin(a3)*cos(c1)
cos(a3)*sin(b2)*cos(c1)+sin(a3)*sin(c1);
sin(a3)*cos(b2) sin(a3)*sin(b2)*sin(c1)+cos(a3)*cos(c1)
sin(a3)*sin(b2)*cos(c1)-cos(a3)*sin(c1);
-sin(b2) cos(b2)*sin(c1) cos(b2)*cos(c1)];
end

```

## A.3 e2R.m

```

function e2R=e2R(e)
% RKM for det kinematiske problem 3-2-1 Eulervinkler
% e : Eulervinkelvektor i grader
% e2R: RKM (3x3)
% Likning: (A-36)-[O-Hallingstad, UNIK 4540- VÅR 2012)
% converts eulervinkler to R
a=e(1)*pi/180;  b=e(2)*pi/180;  c=e(3)*pi/180;

e2R=([cos(a)*cos(b) cos(a)*sin(b)*sin(c)-sin(a)*cos(c)
cos(a)*sin(b)*cos(c)+sin(a)*sin(c);
sin(a)*cos(b) sin(a)*sin(b)*sin(c)+cos(a)*cos(c)
sin(a)*sin(b)*cos(c)-cos(a)*sin(c);
-sin(b) cos(b)*sin(c)
cos(b)*cos(c)]);
end

```

## A.4 S.m

```
function S = S(om)

% Skjevsymmetrisk matrise for om

S = [0 -om(3) om(2);
      om(3) 0 -om(1);
      -om(2) om(1) 0];
end
```

## A.5 R2e.m

```
function R2e = R2e(R)
% det kinematiske problem for 3-2-1 eulervinkler
% Likning: (A-38)-[O-Hallingstad, UNIK 4540-Vår 2012, (A-38)]
% converts R to eulervinkler
R2e=zeros(3,1);
b=atan2(-R(3,1),sqrt(R(1,1)^2+R(2,1)^2));
if 0.8999 < b
    if b < 90.0001
        disp('feil')
    end
end
R2e(1)=atan2(R(3,2)/cos(b),R(3,3)/cos(b));
R2e(3)=atan2(R(2,1)/cos(b),R(1,1)/cos(b));
R2e(2)=b;
R2e=R2e*180/pi;
```

## A.6 kp2dpGa.m

```
function [Ga] = kp2dpGa(F,G,Q,d)
[m, n] = size ( F )
A = [ F G*Q*G' ; zeros(m, n) -F' ] ;
B = expm(A*d) ;
S = B ( 1 : n , n+1: 2* n ) * inv(B( n+1: 2* n , n+1: 2* n ) );
Ga = chol ( S , 'lower');
end
```

## A.7 k2dp.m

```
function [Fi,varargout] = k2dp(d,F,varargin)
% Diskretiserer den kontinuerlige prosesslikninga
% 20121210 Oddvar Hallingstad
% Kall:
% Fi = k2dp(d,F)
% [Fi,La] = k2dp(d,F,L)
% [Fi,Ga] = k2dp(d,F,G,Qb)
% [Fi,La,Ga] = k2dp(d,F,L,G,Qb)
[mF,nF] = size(F);
if mF~=nF, disp('Feil dimensjon på F-matrisa'), end
nvar=length(varargin);
if nvar==0 % d,F --> Fi
    Fi = expm(d*F);
elseif nvar==1 % d,F,L ---> Fi,La
```

```

L = varargin{1};
[mL,nL] = size(L);
if mF~=mL, disp('Feil dimensjon på L-matrisa'), end
A = [ F L ; zeros(nL,nF) zeros(nL,nL)];
B = expm(A*d);
Fi = B(1:mF,1:nF);
varargout{1} = B(1:mF,nF+1:nF+nL);
elseif nvar==2 % d,F,G,Qb ---> Fi,Ga
G = varargin{1}; Qb= varargin{2};
[mG,nG] = size(G); [mQb,nQb] = size(Qb);
if mF~=mG, disp('Feil dimensjon på G-matrisa'), end
if mQb~=nQb, disp('Feil dimensjon på Qb-matrisa'), end
if nG~=mQb, disp('Feil dimensjon på Qb-matrisa'), end
A = [ F G*Qb*G' ; zeros(mF,nF) -F' ] ;
B = expm(A*d);
S = B(1:mF,nF+1:2*nF)*inv(B(mF+1:2*mF ,nF+1:2*nF ));
Fi = B(1:mF,1:nF);
varargout{1} = chol(S,'lower');

% Fi=eye(mF)+d*F;
% Ga=d*G*diag(sqrt(diag(Qb)));
% varargout{1}=Ga;

elseif nvar==3 % d,F,L,G,Qb ---> Fi,La,Ga
L = varargin{1};
[mL,nL] = size(L);
if mF~=mL, disp('Feil dimensjon på L-matrisa'), end
A = [ F L ; zeros(nL,nF) zeros(nL,nL)];
B = expm(A*d);
Fi = B(1:mF,1:nF);
varargout{1} = B(1:mF,nF+1:nF+nL);

G = varargin{2}; Qb= varargin{3};
[mG,nG] = size(G); [mQb,nQb] = size(Qb);
if mF~=mG, disp('Feil dimensjon på G-matrisa'), end
if mQb~=nQb, disp('Feil dimensjon på Qb-matrisa'), end
if nG~=mQb, disp('Feil dimensjon på Qb-matrisa'), end
A = [ F G*Qb*G' ; zeros(mF,nF) -F' ] ;
B = expm(A*d);
S = B(1:mF,nF+1:2*nF)*inv(B(mF+1:2*mF ,nF+1:2*nF ));
varargout{2} = chol(S,'lower');
else
disp('For mange argument i kp2dp')
end
end
end

```

## A.8 kp2dpS.m

```

function [S] = kp2dpS(F,G,Q,dt)
[m, n] = size ( F ) ;
A = [ F G*Q*G' ; zeros(m, n) -F' ] ;
B = expm(A*dt) ;
S = B ( 1 : n , n+1: 2* n ) * inv(B( n+1: 2* n , n+1: 2* n ) ) ;
end

```

## A.9 mprint.m

```
function mprint(A,m,n,s,varargin)
% mprint printer en matrise med det angitte format
% 20121210 Oddvar Hallingstad
% A: todimensional matrise
% m: feltbredde
% n: presisjon
% s: streng, enten 'f', 'e' eller 'g'
% varargin(1): Nummer på startlinjen (nsl)
% varargin(2): Nummer på startkolonnen (nsc)
% varargin(3): Nummer på matrisa (kmtr)
% Eksempler
% m.nf: fast format, n siffer etter punktum
% m.ng: det som er mest kompakt av f eller e, n signifikante siffer
% m.ne: eksponential format, n siffer etter punktum
nvar=length(varargin);
if nvar==0
    nsl=1; nsc=1;
elseif nvar==1
    nsl=varargin(1); nsc=1;
elseif nvar==2
    nsl=varargin{1}; nsc=varargin{2};
elseif nvar==3
    nsl=varargin{1}; nsc=varargin{2}; kmtr=varargin{3};
else
    disp('For mange argument')
    nsl=1; nsc=1;
end
[nl,nc]=size(A);
line=nsl:nsl-1+nl; col=nsc:nsc-1+nc;
str1=strcat('%',num2str(m),'.','0','f'); % Tallformat nummerering
str2=strcat('%',num2str(m),'.',num2str(n),s); % Tallformat matrise
str3=strcat(repmat('#',1,m),repmat(str1,1,nc),'\n'); % Overskrift u/nr
str4=strcat(repmat(str1,1,nc+1),'\n'); % Overskrift m/nr
str5=strcat(str1,repmat(str2,1,nc),'\n'); % Matriselinje
if nvar==3
    fprintf(str4,kmtr,col);
else
    fprintf(str3,col);
end

for k=1:nl
    fprintf(str5,line(k),A(k,:));
end
end
```



